

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfilment of the
requirements for the award of
Bachelor of Technology Degree in Information Technology

By

AMAAN ALI
REG. NO. 40120010



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

**SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI –
600119, TAMILNADU**

APRIL 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND
TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI

CHENNAI-600119

www.sathyabama.ac.in

SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **AMAAN ALI (Reg. No: 40120010)** who carried out the project entitled "**Consolidating Reports Using Observium Tool**" under my supervision from January 2023 to April 2023

Internal Guide

Dr. J. Jabez M.E., Ph.D.,

HEAD OF THE DEPARTMENT

Dr. R. Subhashini M.E., Ph.D.,

Submitted for Viva voce Examination held on 28.04.2023

Internal Examiner

External Examiner

DECLARATION

I, **AMAAN ALI** hereby declare that the project report entitled **Consolidating Reports Using Observium Tool** done by me under the guidance of **Dr. J Jabez** is submitted in partial fulfilment of the requirements for the award of Bachelor of Technology Degree in Information Technology.

DATE: 28.04.2023

PLACE: Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D, Dean, School of Computing** and **Dr. R. Subhashini M.E., Ph.D., Head of the Department of Information Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. J Jabez, M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

CERTIFICATE OF COMPLETION

Ref.No.: 8 /FSPCC/BC001/04/2023



इलेक्ट्रॉनिक्स एवं
सूचना प्रौद्योगिकी मंत्रालय
MINISTRY OF
ELECTRONICS AND
INFORMATION TECHNOLOGY
सत्यमेव जयते



COURSE COMPLETION LETTER

It is certified that

Amaan Ali

Sathyabama Institute of Science and Technology

has successfully undergone the Bridge Course Training Program on

Cloud Computing Technology

Under "FutureSkills PRIME" Programme

commenced from 11 Mar, 2023

Shri. V. Sridharan
Chief Investigator
Cloud Computing Technology
Lead Resource Centre

Cdr.(Retd.) L.R. Prakash
Senior Director
Centre Head
C-DAC, Chennai

ABSTRACT

Network monitoring is the process of constantly monitoring a computer network for problems such as slow traffic or component failure. Network Monitoring tools are always scanning the network and are designed to automatically notify network administrators via text, email, or other application such as Slack when a problem occurs. Network monitoring software differs from network security or intrusion detection systems in that network monitoring is focused on internal network issues such as overloaded routers, server failures, or network connection issues that could impact other devices.

Network monitoring solutions can also initiate failover to remove problem device or circuits from duty until remediation can be performed to repair the issue. Ideally, a proactive network monitoring solution will prevent downtime or failures before they occur by identifying anomalies that could lead to outage if unchecked.

Continuous monitoring is critical to maintaining network integrity. The best network monitoring tools provide visualisation or a dashboard that provides at-a-glance status of monitored network components, indicating any out-of-norm parameters that require further examination or components such as switches, routers, firewalls, servers, and software services, applications, or URLs that could be the source of network disturbances. For maximum effectiveness, a network monitoring system should include high availability components so that a hardware or software failure of systems running the network management tool can be automatically remediated by fail-over to another network monitoring installation.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	i
	LIST OF FIGURES	ii
1	INTRODUCTION	6
	1.1 What is Observium	6
	1.2 What is Network Monitoring	7
	1.3 What is Network Monitoring Tool	8
	1.4 Why is it important?	9
	1.5 Benefits	10
2	AIM AND SCOPE	13
	2.1 Aim	13
	2.2 Scope	13
	2.3 Objectives	14
3	TECHNOLOGICAL OVERVIEW	17
	3.1 Technological Overview	17
4	METHODOLOGY AND ARCHITECTURE	21
	4.1 Methodology	21
	4.2 Architecture	29
5	RESULT AND DISCUSSION	34
	REFERENCES	39

LIST OF FIGURES

FIGURE NO.	FIGURE DESCRIPTION	PAGE NO.
5.1	Front Page	35
5.2	Linux Details	36
5.3	Port Details in Observium	37
5.4	Alert Logging	38
5.5	Alerts	38

CHAPTER 1

INTRODUCTION

1.1 What is Observium?

Observium is a network monitoring and management software that allows users to monitor their IT infrastructure in real-time. It provides comprehensive network monitoring capabilities that can help detect and resolve network issues before they cause downtime or impact the end-users.

Observium supports a wide range of network devices and protocols, including routers, switches, servers, and storage devices. It provides detailed performance metrics, such as CPU usage, memory utilization, network traffic, and system health. Observium is an open-source platform that can be deployed on-premises or in the cloud. It has a user-friendly web interface that displays network performance data in real-time, providing network administrators with the information they need to keep their infrastructure running smoothly.

The main objective of Observium is to provide comprehensive and reliable network monitoring and management capabilities to help network administrators monitor and maintain their IT infrastructure in real-time. Observium aims to provide an easy-to-use, scalable, and customizable platform that can support a wide range of network devices and protocols.

The specific objectives of Observium include:

1. **Monitoring network performance:** Observium collects and analyzes performance metrics for network devices and systems, including CPU usage, memory utilization, network traffic, and system health, among others.
2. **Alerting and notifications:** Observium can be configured to alert network administrators when network issues occur or when performance metrics reach predefined thresholds, allowing them to take action to address the issue promptly.
3. **Capacity planning:** Observium provides historical data on network performance metrics, enabling network administrators to plan for future capacity requirements and optimize network resources effectively.
4. **Troubleshooting:** Observium helps network administrators troubleshoot network issues by providing detailed performance data and real-time insights into network performance, allowing them to identify and resolve issues quickly.

5. Overall, the objective of Observium is to provide a comprehensive, scalable, and easy-to-use platform that enables network administrators to monitor, manage and maintain their IT infrastructure effectively.

1.2 What is a Network Monitoring?

Network monitoring is the process of constantly monitoring a computer network for problems such as slow traffic or component failure. Network Monitoring tools are always scanning the network and are designed to automatically notify network administrators via text, email, or other application such as Slack when a problem occurs. Network monitoring software differs from network security or intrusion detection systems in that network monitoring is focused on internal network issues such as overloaded routers, server failures, or network connection issues that could impact other devices.

Network monitoring solutions can also initiate failover to remove problem device or circuits from duty until remediation can be performed to repair the issue. Ideally, a proactive network monitoring solution will prevent downtime or failures before they occur by identifying anomalies that could lead to outage if unchecked.

Continuous monitoring is critical to maintaining network integrity. The best network monitoring tools provide visualization or a dashboard that provides at-a-glance status of monitored network components, indicating any out-of-norm parameters that require further examination or components such as switches, routers, firewalls, servers, and software services, applications, or URLs that could be the source of network disturbances. For maximum effectiveness, a network monitoring system should include high availability components so that a hardware or software failure of systems running the network management tool can be automatically remediated by fail-over to another network monitoring installation. Network Monitoring should provide:

Visualization of the organization's complete IT and network infrastructure

Monitoring, troubleshooting, and remediation of network performance issues.

Root cause analysis tools when problems occur.

Dashboard with clear visualization tools and reports

The most important benefit of network monitoring is visibility. Having an easy-to-understand, at-a-glance picture of all connected devices across the organization is important, but visualizing the flow of data across devices and networks enables

network administrators to quickly identify and remediate any problems – emergent or potential – before outages occur.

Automating this critical task leads to a higher utilization of critical IT resources. Time not spent putting out network fires can be spent working on projects that create bottom-line value for the organization.

Network monitoring also provides an early warning indication of the utilization of existing infrastructure, giving IT a heads-up indicating the need for upgrading or adding capacity to given network components.

Finally, by comparing traffic to known baselines for a given time of day and season, network monitoring tools can identify unexpected spikes in network traffic that can indicate a problem brewing, whether due to increased demand or cyberattack.

1.3 What is Network Monitoring Tool?

Network monitoring tools gather and analyze network data to provide network administrators with information related to the status of network appliances, link saturation, the most active devices, the structure of network traffic or the sources of network problems and traffic anomalies. Networks are the fundamentals behind businesses worldwide. It plays a pivotal role in serving your employees for administrative purposes and your clients across the continents. The networks help you keep information in a centralized location - accessible to those who need and restrict every other inbound request. So how do you provide continuous top-notch end user experience and maintain your rapidly evolving network? Only by monitoring the availability, health, and performance of your networks over time with the help of reliable, real-time network monitoring tools. Employing individual network monitoring tools for monitoring various network components such as switches/routers, servers, virtual environment, HCI, applications, storage devices, etc., and also employing network monitoring tools for Windows and Linux environments separately is more trouble than help as the tools themselves require constant management, additional resources, and there is also a certain learning curve associated with each network monitoring tool. Therefore, the network performance monitoring tools should support extensive monitoring in a single console. Network scalability is an important aspect to be considered when selecting enterprise network monitoring tools. A network management tool or network monitoring software can be called as scalable when it is more adaptable to the changing needs or demands of the business or users. Scalability helps a network to stay in par with

increased productivity, trends, changing needs and new adaptations. Scalability ensures that the overall network performance does not significantly degrade, even if the size of the network is increasing. Businesses also need remote network monitoring tools for managing multiple geographical locations from one console. In network monitoring, automation helps network performance monitoring tools to react based on threshold values or a set of framed rules/ criteria being met. With automation, the monitoring network tools can automatically detect and troubleshoot problems (proactive monitoring), send alert notifications, forecast storage growth, and much more. When you are monitoring and managing multiple devices in your environment, automation comes in super handy and saves you ample time and resources, making it an important usability aspect of your network monitoring solutions. User Management helps organizations ensure network security by providing access to the designated users only. Apart from providing access to users with roles, the network monitoring tool should also define the scope for users. This helps IT teams with multiple staffs as it clearly defines their operational boundaries. Network monitoring tools with the above features are exceptionally benefiting to your business.

1.4 Why is Network Networking Tool important?

Network failures can impact overall IT performance and cause availability issues across the organization. Network monitoring has several important benefits to the organization by enabling early detection of issues including:

1. Cost savings realized by reducing downtime and speeding remediation by assisting with root cause analysis or displaying network elements that are being over- or under-utilized. Network resources can focus on productive tasks instead of constantly looking for problems.
2. Performance problems can be caught before they impact business operations or lead to a degraded customer experience.
3. Network security enhancements can be realized by detecting unexpected traffic or unknown devices connecting to the network. These could be early indicators of cyberattacks or ransomware attempts.
4. Usage spikes such as logon storms or seasonal traffic jumps can be indicated early on, enabling network administrators to take remedial action to ensure that usage is not impacted.

5. Rogue application usage can be caught. Each business unit may have a group of applications they want tracked and network monitoring can establish which applications and users are doing what on the network.

There are many types of network monitoring. For instance email network monitoring might involve sending test emails and measuring the response time, while web server testing could entail sending an HTTP request to access a given page and log the time until it is served.

First, devices and network connections are identified as are their related performance metrics. Next, the organization determines how frequently to monitor each function. For example, client laptops and printers are not 'network critical' and can have much longer monitoring intervals than routers, switches, and servers that comprise the network backbone. Most network monitoring tools utilize the simple network management protocol (SNMP) to manage and monitor the elements of the network. Most network components are delivered with an SNMP agent which can be used to reconfigure devices, take them offline if they are performing erratically, or to collect information about the device's performance. Network monitoring systems 'ping' the various system ports, and If a device reports a parameter outside of the established threshold an alert is automatically generated so remediation can occur before device failure. Typically, network components are pinged between once a minute and once an hour. Some network devices such as routers and switches utilize the Internet Control Message Protocol (ICMP) to relate information regarding internet protocol (IP) operations and to create error messages when devices fail.

1.5 Benefits of Network Monitoring Tools

1) Network Visibility

You need to be able to monitor every aspect of your network. That includes all of the devices attached to your network and the traffic that travels through the network. It's the best way to keep an eye on the health of your network and identify lags in performance. Just keep tracking of everything on your network can be a challenge. Automated network mapping tools, as part of your network monitoring, can provide a complete view of even the most complex ecosystems.

2) Maintaining Compliance

Organizations that need to maintain compliance with regulations need the appropriate network monitoring tools in place. PCI DSS, HIPAA, FISMA, SOX, and others require network monitoring as part of the internal control structure for

compliance in addition to any external security controls that are already in place.

3) Preventing Downtime

Downtime is a productivity killer and it's expensive. A recent survey reported that 40% of enterprise companies said an hour of downtime would cost between \$1 million and \$5 million. Monitoring can help you prevent unexpected outages. One significant aspect of network monitoring solutions is to identify warning signs that might indicate a device failure or network problem. This helps you identify the problem and prevent downtime from occurring. Not only does network monitoring help prevent downtime but performance monitoring allows IT teams to optimize performance for more efficient operations.

4) Finding and Fixing Problems Quickly

When there is a problem, network monitoring helps you isolate the issue more quickly. Whether it's a traffic fluctuation, a configuration error, or something more serious, network maps can help you quickly find the origin of the problem. Network automation tools, as part of your monitoring solution, can fix many problems automatically. Reducing your Mean Time to Repair (MTTR) reduces the impact of downtime or poor network performance and lets your IT team focus on other issues.

5) Uncovering Security Threats

Network monitoring is primarily used to monitor performance, but it can also help uncover security threats within your system. By continuously monitoring for unusual or suspicious activity, you may be able to detect even small threats before they become big ones. For example, malware or viruses may be undetectable at a glance, but your network monitoring solution can flag unusual activity, such as suspicious use of network resources.

6) Monitoring Bandwidth Utilization

When your bandwidth gets congested, you have unhappy employees and customers. Monitoring bandwidth utilization lets you know when things are started to slow. When bandwidth usage gets close to critical levels, you will be alerted so that you can adjust your quality of service (QoS) protocols or take other actions to improve performance.

7) Capacity Planning

User needs are constantly evolving as well. This can make it difficult to predict how and where users will consume network resources in the future. As utilization increases, it's essential to plan for additional infrastructure and capacity to meet this demand. When you are actively tracking and monitoring performance and

utilization, network monitoring software will help you see when utilization is spiking. By benchmarking current performance, you can more accurately anticipate future capacity and upgrade the network. Historical data can also help make the case for future investments when it's budget time. When you can produce evidence that the network's slowing down or performance is lagging due to increase utilization, it's a lot easier to convince executives that you need to upgrade systems.

8)Deploying New Technologies

Network monitoring is also important when it's time to deploy new technologies. It can help determine if the network can handle additional resources and proactively detect potential performance issues. After deployment, you'll be able to monitor the network to ensure performance doesn't suffer.

9)Freeing Up IT Teams

There's also an undervalued benefit of network monitoring tools. IT teams have a lot on their plates these days. The growth of networks and the complexity has created a significant burden on teams that are typically understaffed, yet expected to ensure perfect performance. The right network monitoring tools, such as WhatsUp Gold, can significantly reduce the burden. IT teams regain control of their network without having to manually dig into performance, get proactive alerts when something needs attention, and reduce the amount of downtime. When a problem is detected, network monitoring solutions make it easier to identify, isolate, and repair the problem. All of this frees up time for team members to focus on other projects that can drive topline growth. That can be invaluable.

10)Report on SLAs

Keeping your promises with regards to network availability is a top priority to IT consultants and managed service providers. With the ability to report on performance that network monitoring provides, you can more easily meet the requirements of SLAs and ensure the satisfaction of your customers

CHAPTER 2

AIM AND SCOPE

2.1 AIM

The main aim of Observium is to provide comprehensive and reliable network monitoring and management capabilities to help network administrators monitor and maintain their IT infrastructure in real-time. Observium aims to provide an easy-to-use, scalable, and customizable platform that can support a wide range of network devices and protocols. By using we can consolidate the reports of the networks and manage it properly. Consolidating reports is to provide a unified view of data and information from multiple sources in a single report or dashboard. Consolidation of reports helps organizations to streamline their reporting process, improve decision-making, and gain insights into their business performance across different departments, systems, and locations.

Overall, the objective of consolidating reports using Observium network monitoring tool is to provide organizations with a single, comprehensive view of their business performance, enabling them to make informed decisions and drive improvements across their operations.

2.2 SCOPE

The scope of a proposed network management is variable, since variables are the elements to monitor in each case and, moreover, is a living project that evolves at the same time as the network infrastructure evolves. Also, as you reach milestones set at the beginning of the project will set new goals so that every time we get more control and to better manage on our platform.

As we there is a steep increase in the dependency of internet and networking and emerging technologies like cloud demands the need for network speed and high availability of network, there is a great scope for a network professionals in India.

Also all the mobile service providers and home, business internet service providers need network professionals

2.3 OBJECTIVES

The main objective of Observium is to provide comprehensive and reliable network monitoring and management capabilities to help network administrators monitor and maintain their IT infrastructure in real-time. Observium aims to provide an easy-to-use, scalable, and customizable platform that can support a wide range of network devices and protocols.

Observium collects and analyzes performance metrics for network devices and systems, including CPU usage, memory utilization, network traffic, and system health, among others.

Network performance monitoring aids network administrators and analysts in gathering network data, allowing them to measure performance variables and identify potential issues or risks. Network performance monitoring tools will help in handling the management between linked devices, as well as gathering metrics from servers or other linked devices. After this, a network performance monitoring tool will analyze the collected data to find any bottlenecks or congestion points, so the network can increase its throughput once fixed. Some of these tools can be automated to make the process easier. These tools can create reports and alert users when any issues appear. Ideally, this will allow network admins to address issues before they become bigger problems.

Network performance monitoring is an important general IT process that allows administrators to optimize a network's performance -- especially in organizations with strict reliability requirements and in network architectures that are complex. Networking components such as servers, routers, switches and virtual machines (VMs) can be monitored for performance.

If there are bottlenecks in a network, then user experience will be negatively affected. This is why network performance monitoring must consider what the user experience will be like by monitoring the network for bottlenecks or congestion and providing better throughput.

Alerting and notifications: Observium can be configured to alert network administrators when network issues occur or when performance metrics reach predefined thresholds, allowing them to take action to address the issue promptly.

Capacity planning: Observium provides historical data on network performance metrics, enabling network administrators to plan for future capacity requirements and optimize network resources effectively.

Troubleshooting: Observium helps network administrators troubleshoot network

issues by providing detailed performance data and real-time insights into network performance, allowing them to identify and resolve issues quickly.

Another group of users are security admins. Historically, they would be relying on different sets of tools - firewalls, endpoint protection, and incident management systems. It turns out, however, that the network perspective is also an excellent way to discover ongoing compromise and hunt down threat actors operating within the perimeter.

Vendors have therefore invested a great deal of effort into developing the network-centric approach to security, giving rise to an area of the market known as NDR, or Network Detection and Response, which is, in essence, the materialization of a detect-and-respond mindset (next to the traditional practice of prevent and protect). Observium gathers data from devices with the help of SNMP and display those data in graphical pattern via a web interface. It makes hefty use of the RRDtool package. It has a number of thin core design goals, which includes collecting as much historical information about devices, being totally auto-discovered with slight or no manual interruption, and having a very simple yet powerful interface.

Traffic Accounting is a feature available in the Observium which is designed to ease the process of tracking and billing customer bandwidth usage. It independently polls and stores measurements in its own database tables to bypass traditional round-robin limitations. It can be run at different intervals to the primary Observium poller allowing accounting to be based on intervals other than the default 5 minutes.

Observium's threshold alerting system allows you to configure thresholds and failure states for a wide range of different entity types. Some types support automatic collection of thresholds from the monitored device itself.

Alerting entries are automatically created during the discovery process making sure that your alerting system is kept up to date with the configuration of your network infrastructure, helping to reduce the number of missed alerts.

The objective of consolidating a report is to compile and integrate data from various sources into a single document, ensuring accuracy and consistency. By consolidating a report, the aim is to create a comprehensive overview of the data, which can be analyzed and used for informed decision making. The specific objectives for consolidating a report may include:

Data Accuracy: Ensuring that all data included in the report is accurate and reliable, by cross-checking and verifying information from multiple sources.

Data Consistency: Ensuring that data is presented in a consistent manner

throughout the report, with standardized formats, units, and terminology, to facilitate easy comparison and analysis.

Data Integration: Consolidating data from different sources, such as databases, spreadsheets, or other reports, into a single document, to provide a holistic view of the information and identify patterns or trends.

Data Analysis: Preparing the consolidated report in a way that enables effective data analysis, by organizing and summarizing data in a meaningful and insightful manner, such as through charts, graphs, or tables.

Decision Making: Providing a reliable and comprehensive report that can be used as a basis for informed decision making, by presenting relevant data in a clear and concise manner, and highlighting key findings or insights.

Efficiency: Streamlining the process of consolidating the report to ensure that it is done efficiently and effectively, by optimizing data collection, validation, and presentation methods.

By achieving these objectives, the consolidated report can serve as a valuable tool for stakeholders to make informed decisions, identify opportunities, and address challenges based on accurate and reliable data. Consolidated report is to present a comprehensive view of a company's financial performance and position by combining the financial information of multiple entities within the organization. The report aims to provide stakeholders with a clear understanding of the company's overall financial health and to identify trends, opportunities, and challenges that may impact its future performance.

Consolidated reports typically include financial statements such as balance sheets, income statements, and cash flow statements, as well as footnotes and other supplementary information that provide context and detail on the financial data presented. The report may also include management commentary and analysis that explains the key drivers of performance and the company's strategic priorities going forward. The objective of a consolidated report is to provide stakeholders with a transparent and accurate picture of a company's financial position and performance, enabling them to make informed decisions about investing, lending, or partnering with the company.

CHAPTER 3

TECHNOLOGICAL OVERVIEW

Observium is a network monitoring and management software that provides comprehensive visibility into network devices and their performance. It offers a wide range of features for monitoring and managing networks, including automated device discovery, performance monitoring, event management, reporting, configuration management, device support, extensibility, and community support.

Technological overview of Observium:

1. Auto-discovery: Observium has an auto-discovery feature that can automatically identify and add network devices to its monitoring system. It uses protocols such as SNMP, ICMP, and IPMI to discover devices on the network, reducing manual effort and ensuring that all devices are monitored.
2. Performance Monitoring: Observium collects performance metrics from network devices using protocols like SNMP, ICMP, IPMI, and others. It monitors various parameters such as CPU usage, memory utilization, interface traffic, errors, and more, providing real-time visibility into the health and performance of the network.
3. Event Management: Observium allows users to configure alerts based on predefined or custom thresholds for specific metrics or events. When a threshold is breached, Observium can send notifications through email, SMS, or other channels, allowing users to quickly identify and resolve network issues.
4. Graphical Reporting: Observium provides graphical reporting options, including customizable graphs, charts, and tables, that allow users to visualize network performance data over time. This helps in identifying trends, patterns, and anomalies in the network's performance.
5. Configuration Management: Observium has configuration management capabilities that allow users to manage and track changes in network device configurations. It provides configuration backup, comparison, and change tracking features, helping users ensure that devices are configured correctly and in compliance with established standards.
6. Device Support: Observium supports a wide range of network devices, including routers, switches, firewalls, servers, wireless access points, and more. It also supports various protocols such as SNMP, ICMP, IPMI, JFlow, sFlow, NetFlow, and more, making it versatile for monitoring diverse network environments.

7. **Extensibility:** Observium is highly extensible and supports custom scripts, plugins, and modules, allowing users to add additional functionality or integrate with other systems or tools as needed. This provides flexibility in extending the capabilities of Observium to suit specific monitoring requirements.
8. **Community Support:** Observium has an active community of users and contributors who provide support, documentation, and community-driven updates. The community support includes forums, wikis, mailing lists, and other resources that help users with troubleshooting, customization, and best practices.
9. **User Interface:** Observium provides a web-based interface that is known for its simplicity and ease of use. The interface is designed to provide a user-friendly experience, making it easy to navigate, configure, and monitor the network.
10. **Architecture:** Observium is built on a LAMP (Linux, Apache, MySQL, PHP) stack, which is a popular open-source technology stack for web-based applications. It utilizes PHP for backend processing, MySQL for data storage, and Apache for web server functionality.
11. **Data Collection:** Observium collects data from network devices using protocols like SNMP, ICMP, and IPMI. This data includes performance metrics such as CPU usage, memory utilization, interface traffic, and more. Observium also collects configuration data and logs from devices, allowing users to monitor changes and troubleshoot issues.
12. **Data Storage:** Observium stores collected data in a database, allowing users to access and analyze data over time. Observium uses MySQL, PostgreSQL, or MariaDB as its database backend.
13. **Web-Based Interface:** Observium has a web-based interface that allows users to view and analyze collected data. The interface uses HTML, CSS, and JavaScript technologies to provide a user-friendly and responsive experience across devices.
14. **Alerting and Notifications:** Observium provides alerting and notification capabilities that use technologies like email, SMS, and push notifications to inform users of threshold breaches or other events.
15. **Graphing and Reporting:** Observium uses technologies like RRDtool and Highcharts to provide customizable and interactive graphs and reports of collected data, allowing users to identify trends and patterns in network performance.
16. **Extensibility:** Observium is highly extensible and supports custom scripts, plugins, and modules. These can be written in various programming languages, including PHP, Perl, Python, and Bash, allowing users to add additional functionality

or integrate with other systems or tools as needed.

17. Device Support: Observium supports a wide range of network devices, including routers, switches, firewalls, servers, wireless access points, and more. It also supports various protocols such as SNMP, ICMP, IPMI, and more, making it versatile for monitoring diverse network environments.

18. Community Support: Observium has an active community of users and contributors who provide support, documentation, and community-driven updates. The community supports various technologies and tools used in conjunction with Observium, including Docker, Ansible, and more.

Observium is a powerful network monitoring and management solution that offers a wide range of features for monitoring, managing, and troubleshooting networks. It is widely used in various industries, including IT, telecommunications, data centers, and more, to gain visibility and control over network devices and ensure optimal network performance. Observium uses a combination of data collection, storage, visualization, and alerting technologies to provide comprehensive network monitoring and management capabilities. Its extensibility and community support make it a popular choice for network administrators and engineers who require a customizable and flexible monitoring solution. Observium provides a robust and extensible solution for network monitoring and management, with data collection, storage, display, alerting, and event management capabilities. It supports a wide range of network devices, protocols, and vendors, and has an active community of users and contributors for support and updates. It's important to consult the official documentation, user guides, and community resources for detailed information on the specific version and configuration of Observium being used.

The objective for a consolidate report is to gather information from multiple sources and present it in a clear, concise, and organized manner. The report should provide a comprehensive overview of the topic at hand, with all relevant information presented in a way that is easy to understand and analyze. The purpose of the report may vary depending on the audience and the specific goals of the project, but it generally aims to provide a summary of the relevant data, insights, and recommendations related to the topic. Ultimately, the objective of a consolidate report is to help the reader make informed decisions based on the information presented. To create a comprehensive and organized report by consolidating relevant data and information from multiple sources, in order to provide a clear and concise overview of the subject matter. The report aims to provide accurate,

reliable, and relevant information that can be easily understood and used for decision-making, problem-solving, or informational purposes. The consolidation of data and information will ensure that the report is comprehensive, reliable, and presents a holistic view of the topic, enabling the intended audience to gain valuable insights and make informed decisions based on the findings.

The objective of preparing a consolidated report is to provide a comprehensive overview of various data, information, or findings from multiple sources or departments. Consolidated report aims to synthesize and summarize relevant details in a clear and organized manner, presenting a holistic view of the subject matter. The key objectives of a consolidated report may include:

Centralizing Information: The consolidated report aims to gather data or insights from different sources or departments and present them in a centralized format. It helps to streamline information and make it easily accessible for decision-making or analysis purposes.

Providing an Overview: The consolidated report should provide a concise and informative overview of the key findings, results, or conclusions from the various sources. It should present a high-level summary that highlights the main points or trends, enabling readers to grasp the main message quickly.

Identifying Patterns or Trends: A consolidated report may aim to identify patterns, trends, or correlations across different datasets or information sources. It may involve analyzing data, identifying commonalities or trends, and drawing meaningful conclusions that can inform decision-making or strategic planning.

Synthesizing Insights: The consolidated report may aim to synthesize insights from multiple sources or perspectives to provide a well-rounded understanding of the topic. It may involve combining data, analysis, or opinions from various stakeholders to generate comprehensive insights or recommendations.

Enhancing Communication: The consolidated report should aim to improve communication by presenting complex or diverse information in a concise and easily understandable format. It should use clear language, visual aids, or other communication techniques to ensure that the report is accessible and meaningful to the intended audience.

Supporting Decision-making: The consolidated report may aim to provide relevant information or insights that can support decision-making processes. It should present reliable data, analysis, or recommendations that can inform strategic decisions, operational planning, or policy formulation.

CHAPTER 4

METHODOLOGY AND ARCHITECTURE

4.1 METHODOLOGY

4.1.1 Poller:

In Observium, a poller refers to a component that is responsible for polling network devices and collecting data from them. Observium uses a polling mechanism to periodically query devices using various protocols, such as SNMP (Simple Network Management Protocol), ICMP (Internet Control Message Protocol), and other methods, to collect data about the devices' status, performance metrics, and other information.

Here are the key aspects of pollers in Observium:

1. **Polling Interval:** Observium allows you to configure the polling interval, which determines how frequently devices are polled for data. The polling interval can be set globally or on a per-device basis, allowing you to control the frequency of data collection based on your monitoring requirements and the devices' characteristics.
2. **Data Collection:** Pollers in Observium collect data from devices using various protocols, such as SNMP for collecting performance metrics like CPU utilization, memory usage, and interface traffic, ICMP for checking device availability and latency, and other methods for gathering data from different types of devices.
3. **Data Storage:** Pollers in Observium store the collected data in a database, allowing you to retain historical data for trend analysis, reporting, and troubleshooting purposes. The data is stored in a structured format, which allows you to query and analyze the data using Observium's built-in reporting and visualization features or external tools.
4. **Device Polling Settings:** Observium allows you to configure various polling settings for each device, such as SNMP versions, community strings, polling timeouts, and retries. These settings allow you to customize the polling behavior for different devices based on their configurations and requirements.
5. **Poller Performance:** Pollers in Observium are designed to be efficient and scalable, with optimizations for handling large numbers of devices and collecting data from them in parallel. Observium provides performance monitoring and tuning options for optimizing the poller performance, such as adjusting the number of concurrent threads, optimizing database queries, and tuning polling intervals, to

ensure efficient data collection and processing.

6. Failover and Redundancy: Observium provides failover and redundancy mechanisms for pollers to ensure high availability of monitoring services. If a poller becomes unavailable, devices assigned to that poller can be automatically reassigned to other available pollers, ensuring continuous data collection and monitoring.

Pollers in Observium are a critical component of the monitoring infrastructure, responsible for collecting data from devices and populating the database with the collected data. They allow Observium to provide real-time monitoring, historical data analysis, and reporting for network devices, helping network administrators to effectively manage and troubleshoot their networks.

4.1.2 Database:

Observium uses a relational database to store the collected data from monitored devices. The type of database used by Observium can be configured during the installation process, and the supported databases include:

MySQL: MySQL is an open-source relational database management system (RDBMS) that is commonly used with Observium. It is known for its performance, reliability, and ease of use, making it a popular choice for storing Observium's monitoring data.

MariaDB: MariaDB is a community-developed, open-source fork of MySQL that is fully compatible with MySQL and can be used interchangeably with MySQL in Observium. It offers similar features and performance characteristics, making it a viable alternative to MySQL.

PostgreSQL: PostgreSQL is another open-source relational database management system that can be used with Observium. It is known for its advanced features, including support for complex data types, extensibility, and robustness.

Observium uses the selected database to store various types of data collected from monitored devices, including performance metrics, health status, configuration information, and historical data. The data is organized into tables and stored in a structured format, allowing for efficient retrieval and analysis through queries and other database operations. It's important to note that Observium includes an automated installation process that sets up the required database schema and tables during installation. Users do not need to manually create the database schema, as Observium takes care of this as part of the installation process.

Additionally, Observium provides tools for managing and maintaining the database, such as database backups, optimizations, and upgrades, to ensure smooth operation and data integrity.

4.1.3 Web User Interface:

Observium provides a web-based user interface (UI) that allows users to access the collected data in a visually appealing and user-friendly way. The web UI provides a wide range of features and functionalities for monitoring and managing network devices. Here are some key aspects of the web UI in Observium:

1. **Dashboard:** The dashboard is the landing page of the Observium web UI and provides an overview of the monitored devices and their status. It displays various widgets, such as graphs, tables, and maps, that provide a snapshot of the network health and performance. Users can customize the dashboard to include the widgets that are most relevant to their monitoring needs.
2. **Device View:** Observium provides detailed device views that display comprehensive information about individual devices being monitored. Device views include graphs, tables, and other visualizations that show performance metrics, health status, and configuration details of the device. Users can navigate through different tabs in the device view to access specific types of data, such as interfaces, sensors, storage, and more.
3. **Graphs and Charts:** Observium offers a rich set of graphs and charts that visualize network performance data. Users can view historical data, monitor trends, and analyze network performance using various types of graphs, such as line graphs, bar charts, and pie charts. Observium also supports zooming, panning, and customization of graph settings to suit specific monitoring requirements.
4. **Reports:** Observium allows users to generate reports based on the collected data. Reports can include information such as device availability, performance trends, and historical data. Reports can be generated in various formats, such as PDF, CSV, and XML, and can be scheduled for automatic generation and delivery via email.
5. **Alerts and Notifications:** Observium provides a system for defining alerts and notifications based on predefined thresholds. Users can configure alert rules to trigger notifications, such as email or SMS, when network devices exceed specified thresholds or when specific events occur. Alerts and notifications help network administrators quickly identify and address network issues.

6. Search and Filtering: Observium includes powerful search and filtering capabilities that allow users to quickly locate and filter devices, interfaces, sensors, and other data based on various criteria, such as device name, IP address, location, and more. This makes it easy to find and focus on specific devices or data of interest.

7. Customization: Observium provides customization options that allow users to configure the web UI to suit their preferences and requirements. Users can customize the layout, appearance, and behavior of the web UI, as well as define custom dashboards, views, and reports.

Overall, the web UI in Observium is designed to provide a user-friendly and intuitive interface for monitoring and managing network devices, with a wide range of features and functionalities to visualize, analyze, and act upon the collected data.

4.1.4 Collection Modules

Observium uses various collection modules to collect data from different types of devices and systems. These collection modules are responsible for gathering performance metrics, health status, configuration information, and other relevant data from the devices being monitored. Here are some commonly used collection modules in Observium:

1. SNMP (Simple Network Management Protocol): SNMP is a widely used network management protocol that allows Observium to collect data from network devices that support SNMP. Observium uses SNMP to gather performance metrics, interface status, CPU and memory utilization, and other device-specific data from devices such as switches, routers, servers, and printers.

2. Agent-based Collection: Observium supports agent-based collection, which involves installing an agent software on the devices being monitored. The agent software collects data locally from the device and sends it to Observium for processing. This allows for more detailed and comprehensive data collection, as the agent can collect data that may not be available via SNMP. Observium supports various agent-based collection protocols, such as SNMP agent, Syslog, NRPE (Nagios Remote Plugin Executor), and SNMP traps.

3. Syslog Collection: Observium can collect syslog data from devices that send syslog messages. Syslog is a standard protocol used for sending log messages from network devices, such as routers, switches, firewalls, and servers. Observium

can parse and process syslog messages to extract relevant information, such as system events, errors, and notifications.

4. API Collection: Observium can collect data from devices and systems that provide APIs (Application Programming Interfaces) for data retrieval. This allows for custom integration with third-party systems and devices that may not support SNMP or other standard protocols. Observium supports various API protocols, such as RESTful APIs, XML APIs, and JSON APIs.

5. WMI (Windows Management Instrumentation): Observium can collect data from Windows-based devices using WMI, which is a set of extensions to the Windows operating system that provides management information and control of devices and applications. Observium can gather performance metrics, system information, and configuration data from Windows-based devices using WMI.

6. J-Flow, sFlow, NetFlow, and other flow-based collection: Observium can collect network flow data from devices that support flow-based protocols, such as J-Flow, sFlow, NetFlow, and other similar protocols. Flow data provides information about network traffic patterns, such as source and destination IP addresses, ports, protocols, and other relevant data for network traffic analysis.

7. Custom Scripts: Observium allows users to create custom scripts or plugins to collect data from devices that do not support standard protocols. Custom scripts can be written in various scripting languages, such as Bash, Perl, Python, or PHP, and can be executed by Observium to collect data from devices using custom methods or APIs.

These are some of the collection modules used in Observium to gather data from different types of devices and systems. Observium provides flexibility in choosing and configuring the appropriate collection modules based on the devices being monitored and the data that needs to be collected for effective network monitoring and management.

4.1.5 Alerting

Observium provides robust alerting capabilities that allow you to configure and customize alerts based on various conditions and thresholds. When these conditions are met, Observium can send notifications to designated recipients via email, SMS, or other methods. Here are the key components of alerting in Observium:

1. **Alert Rules:** Observium allows you to define alert rules, which are conditions or thresholds that trigger an alert. Alert rules can be configured based on various parameters, such as device status, interface utilization, CPU or memory utilization, packet loss, latency, and other performance metrics. Alert rules can be configured globally or on a per-device or per-interface basis, providing flexibility in setting up specific alerts for different devices or interfaces.
2. **Notification Contacts:** Observium allows you to define notification contacts, which are the recipients of alert notifications. Notification contacts can be configured with different contact methods, such as email, SMS, or custom scripts. You can configure multiple notification contacts and group them to receive different types of alerts, allowing you to customize notifications based on severity or importance.
3. **Notification Methods:** Observium supports various notification methods, including email, SMS, custom scripts, and third-party integrations. You can configure the desired notification methods for each notification contact, and Observium will use these methods to send notifications when alerts are triggered.
4. **Escalation Rules:** Observium allows you to define escalation rules, which specify the actions to be taken if an alert is not acknowledged or resolved within a certain timeframe. Escalation rules can be configured to escalate alerts to different notification contacts, change the severity level of alerts, or trigger additional actions, such as executing custom scripts or sending notifications to higher-level contacts.
5. **Custom Alert Templates:** Observium allows you to define custom alert templates, which are customizable alert messages that can be sent to notification contacts. You can define custom variables in alert templates to include dynamic information, such as device name, interface name, or performance metrics, in the alert notifications, providing more context and actionable information in the alerts.
6. **Alert Suppression:** Observium provides alert suppression capabilities, which allow you to configure suppression rules for specific devices, interfaces, or performance metrics. This can help to reduce unnecessary alerts and prevent alert storms during maintenance windows or known periods of high utilization.
7. **Alert History and Acknowledgment:** Observium maintains an alert history, which allows you to view the history of triggered alerts, acknowledged alerts, and resolved alerts. You can acknowledge alerts to indicate that they have been addressed or add notes to provide additional information about the alert resolution.

Observium's alerting capabilities provide a flexible and powerful mechanism for monitoring and alerting on network devices and systems. With its customizable alert

rules, notification contacts, escalation rules, and other features, Observium allows you to configure alerts that are tailored to your specific monitoring requirements and helps you proactively identify and address network issues.

4.1.6 Event Handler

Event Handlers in Observium are custom scripts or commands that can be executed automatically in response to specific events or alerts triggered by Observium. Event Handlers provide a way to automate actions or perform additional tasks based on the events detected by Observium, allowing for customized actions and integrations with other systems. Here are the key components of Event Handlers in Observium:

1. **Event Rule:** Observium allows you to define Event Rules, which are conditions or events that trigger an Event Handler. Event Rules can be based on various parameters, such as device status, interface utilization, CPU or memory utilization, packet loss, latency, and other performance metrics. Event Rules can be configured globally or on a per-device or per-interface basis, providing flexibility in setting up specific actions for different devices or interfaces.
2. **Event Handler:** Observium allows you to define custom Event Handlers, which are scripts or commands that are executed automatically in response to events triggered by Event Rules. Event Handlers can be written in any scripting language that is supported by the operating system where Observium is installed, such as Bash, Python, Perl, or PHP. Event Handlers can perform a wide range of actions, such as sending notifications, executing commands on remote devices, updating databases, sending data to external systems, or performing other custom tasks.
3. **Event Handler Execution:** When an Event Rule is triggered, Observium automatically executes the configured Event Handler associated with that rule. The Event Handler receives information about the triggered event as parameters or environmental variables, allowing it to process the event and take appropriate actions.
4. **Customization:** Event Handlers in Observium provide a high degree of customization. You can create your own scripts or commands that are tailored to your specific requirements and integrate with other systems or tools in your environment. You can also use the available variables and data passed to the Event Handlers to dynamically generate actions or notifications based on the specific

event triggered.

5. Logging and Debugging: Observium logs the execution of Event Handlers, allowing you to track the actions performed by the Event Handlers and troubleshoot any issues. Observium also provides options for debugging and testing Event Handlers, allowing you to validate their functionality before deploying them in a production environment.

Event Handlers in Observium provide a powerful mechanism for automating actions and integrations based on events or alerts triggered by Observium. They enable you to customize and extend the functionality of Observium to meet your specific monitoring and automation requirements, making Observium a versatile and flexible network monitoring solution.

4.1.7 Distributed Polling

Observium, by default, uses a central polling architecture where a single Observium instance polls all the devices and collects data from them. However, starting from Observium version 21.1, Observium also supports distributed polling, which allows you to distribute the polling load across multiple Observium instances. Here are the key components of distributed polling in Observium:

1. Pollers: In a distributed polling setup, you can configure multiple Observium instances as pollers. Each poller is responsible for polling a subset of devices and collecting data from them. Pollers can be deployed on separate physical or virtual machines, or on different network segments to distribute the polling load.
2. Poller Configuration: Each poller needs to be configured with its own Observium configuration, including database settings, device polling settings, and other configuration parameters. This allows each poller to operate independently and poll devices based on its own configuration.
3. Poller Coordination: Observium uses a distributed coordination mechanism to ensure that the polling load is evenly distributed across the pollers. This coordination is done through a shared database or a central coordination server, which allows pollers to communicate with each other and synchronize their polling activities.
4. Device Assignment: Devices in Observium can be assigned to specific pollers or groups of pollers based on various criteria, such as device type, location, or custom attributes. This allows you to control which poller is responsible for polling which devices, and ensures that the polling load is distributed based on your

requirements.

5. **Failover and Redundancy:** Observium provides failover and redundancy mechanisms in a distributed polling setup. If a poller becomes unavailable, devices assigned to that poller can be automatically reassigned to other available pollers. This ensures that polling continues uninterrupted even in the case of poller failures.

6. **Central Instance:** In a distributed polling setup, you can still have a central Observium instance that serves as the main interface for managing and monitoring your network. The central instance can aggregate data from all the pollers and provide a unified view of the network status and performance.

7. **Scalability:** Distributed polling in Observium allows you to scale your monitoring infrastructure horizontally by adding more pollers as needed. This helps to distribute the polling load and accommodate growing networks with a large number of devices.

Distributed polling in Observium provides a scalable and flexible solution for managing and monitoring large networks with distributed devices. It allows you to distribute the polling load, improve performance, and ensure high availability of monitoring services. However, setting up and managing a distributed polling architecture requires careful planning, configuration, and coordination among the pollers, and may require additional infrastructure and resources to support the distributed setup.

4.2 ARCHITECTURE

The architecture of Observium is based on a client-server model, where the Observium server collects and manages data from network devices, and clients interact with the server to access monitoring data through the web user interface or APIs. Here are the key components of the Observium architecture:

4.2.1 Observium Server

Observium Server is the core component of the Observium architecture and is responsible for collecting, processing, and storing monitoring data from network devices. The Observium server collects data from network devices using various protocols such as SNMP, ICMP, and others, to collect data about device status, performance metrics, and other information. The server uses collectors, such as the default poller, distributed pollers, and pollers for specific protocols or device types, to collect the data and populate the database with the collected data.

The Observium server stores the collected data in a database, which serves as the central repository for all the monitoring data. Observium uses a relational database management system (RDBMS), such as MySQL, PostgreSQL, or MariaDB, to store the data in a structured format for efficient querying, analysis, and reporting. It provides a web-based user interface that allows users to interact with the server and access monitoring data. The web user interface provides various features for configuring devices, viewing device status, performance metrics, and other information, setting up alerting, managing events, and generating reports.

Observium server provides APIs (Application Programming Interfaces) that allow clients to interact with the Observium server programmatically. The APIs provide a means for integrating Observium with other systems, automating tasks, and extracting monitoring data for custom applications or scripts. The Observium server architecture is designed to be scalable, allowing you to add more devices and collectors as your network grows. You can also use distributed pollers to distribute the load across multiple servers for better performance and scalability. The Observium server provides security features, such as user authentication and authorization, to control access to monitoring data. You can also configure SSL/TLS encryption to secure data communication between the server and clients.

Overall, the Observium server is a powerful and flexible tool for monitoring network devices. Its architecture is designed to be scalable, extensible, and secure, providing a comprehensive monitoring solution for network administrators.

4.2.2 Observium Database

The Observium database is the central repository for all the monitoring data collected from network devices by the Observium server. The database stores the data in a structured format, allowing for efficient querying, analysis, and reporting. Observium uses a relational database management system (RDBMS) to store the data, and supports several popular RDBMSs such as MySQL, PostgreSQL, and MariaDB.

The Observium database schema consists of multiple tables that store different types of data, such as device information, interface data, port data, and more. Some of the key tables in the Observium database schema are:

1. **devices:** This table stores information about the network devices being monitored, such as their hostname, IP address, device type, and SNMP community string.

2. ports: This table stores information about the network ports on the devices being monitored, such as their speed, duplex mode, and link status.
3. interfaces: This table stores information about the network interfaces on the devices being monitored, such as their operational status, traffic statistics, and errors.
4. sensors: This table stores information about various sensors on the devices being monitored, such as temperature sensors, fan speed sensors, and power supply sensors.
5. syslogs: This table stores syslog messages received from the devices being monitored.
6. alerts: This table stores information about alerts generated by the Observium server, such as the alert type, severity, and time stamp.

The Observium database can be accessed using SQL queries or through the Observium web user interface. The web user interface provides various features for querying and analyzing the monitoring data, such as device status, interface traffic, and alert history. The database can also be queried programmatically using the Observium API.

4.2.3 Observium Web Interface and APIs

Observium provides a web-based user interface that allows users to interact with the Observium server and access monitoring data. The web user interface provides a comprehensive set of features for configuring devices, viewing device status, performance metrics, and other information, setting up alerting, managing events, and generating reports. Some key features of the Observium web user interface include:

1. Device Dashboard: The device dashboard provides an overview of the status and performance of a specific device being monitored. It includes information such as device uptime, CPU usage, memory usage, interface status, and other important metrics in a graphical format for easy visualization.
2. Device Inventory: The device inventory feature allows users to manage and view detailed information about all the devices being monitored. It provides options for organizing devices into groups, assigning custom properties, and managing device configurations.
3. Interface Monitoring: Observium provides detailed monitoring of network interfaces, allowing users to view interface status, traffic statistics, errors, and other

performance metrics. It provides graphical representations of interface data for easy visualization and analysis.

4. Alerting and Event Management: Observium allows users to configure alerts based on various criteria such as device status, performance thresholds, and syslog messages. Alerts can be configured to trigger notifications via email, SMS, or other methods. The web user interface also provides features for managing events, acknowledging alerts, and resolving issues.

5. Reports and Graphs: Observium provides a wide range of pre-configured reports and graphs for analyzing monitoring data. Users can generate reports and graphs for various performance metrics, interface data, and other data points, and customize them according to their requirements.

Observium also provides an Application Programming Interface (API) that allows clients to interact with the Observium server programmatically. The API provides a means for integrating Observium with other systems, automating tasks, and extracting monitoring data for custom applications or scripts. The Observium API supports RESTful API calls and provides access to various functionalities, including device management, polling data retrieval, alert management, and event management. The combination of the Observium web user interface and API provides a powerful and flexible solution for monitoring and managing network devices, allowing users to configure, monitor, and analyze network devices in a convenient and efficient manner.

4.2.4 Scalability and Security

Observium is designed to be scalable and secure, making it suitable for monitoring large networks and ensuring the confidentiality, integrity, and availability of monitoring data.

In terms of scalability, Observium is built on a distributed architecture that allows for horizontal scaling, meaning that multiple Observium pollers can be deployed to distribute the polling load across multiple servers. This allows for efficient monitoring of a large number of devices and interfaces without overloading a single server. Additionally, Observium uses a multi-threaded polling engine that can handle concurrent polling of multiple devices, further enhancing its scalability.

Observium also supports distributed polling, where pollers can be deployed in different geographic locations to monitor devices across multiple sites or data centres. This allows for efficient monitoring of geographically dispersed networks

and reduces the latency of polling data retrieval.

When it comes to security, Observium provides various features to protect the monitoring data and ensure the integrity of the system. This includes support for encrypted communication using SSL/TLS for secure communication between the Observium server, pollers, and clients. Observium also supports role-based access control (RBAC) that allows for granular control over user permissions and access to monitoring data, ensuring that only authorized users can access and modify the monitoring settings and data. Observium also has built-in security features such as support for SNMPv3, which provides enhanced security features such as authentication and encryption for SNMP communication. Observium also supports SNMP traps for receiving notifications from devices in a secure manner.

In addition, Observium provides options for data retention and archiving, allowing for long-term storage of monitoring data for compliance and historical analysis purposes. This helps ensure data integrity and availability even over extended periods of time. Observium also actively maintains and releases regular updates and security patches to address known vulnerabilities and ensure the security of the system.

In summary, Observium is designed to be scalable and secure, providing efficient and reliable monitoring of large networks while ensuring the confidentiality, integrity, and availability of monitoring data.

CHAPTER 5

RESULT AND OBSERVATION

Results and Observation of project consolidation of reports using observium:

Observium is designed to be scalable and secure, making it suitable for monitoring large networks and ensuring the confidentiality, integrity, and availability of monitoring data.

In terms of scalability, Observium is built on a distributed architecture that allows for horizontal scaling, meaning that multiple Observium pollers can be deployed to distribute the polling load across multiple servers. This allows for efficient monitoring of a large number of devices and interfaces without overloading a single server. Additionally, Observium uses a multi-threaded polling engine that can handle concurrent polling of multiple devices, further enhancing its scalability.

Observium also supports distributed polling, where pollers can be deployed in different geographic locations to monitor devices across multiple sites or data centers. This allows for efficient monitoring of geographically dispersed networks and reduces the latency of polling data retrieval.

When it comes to security, Observium provides various features to protect the monitoring data and ensure the integrity of the system. This includes support for encrypted communication using SSL/TLS for secure communication between the Observium server, pollers, and clients. Observium also supports role-based access control (RBAC) that allows for granular control over user permissions and access to monitoring data, ensuring that only authorized users can access and modify the monitoring settings and data.

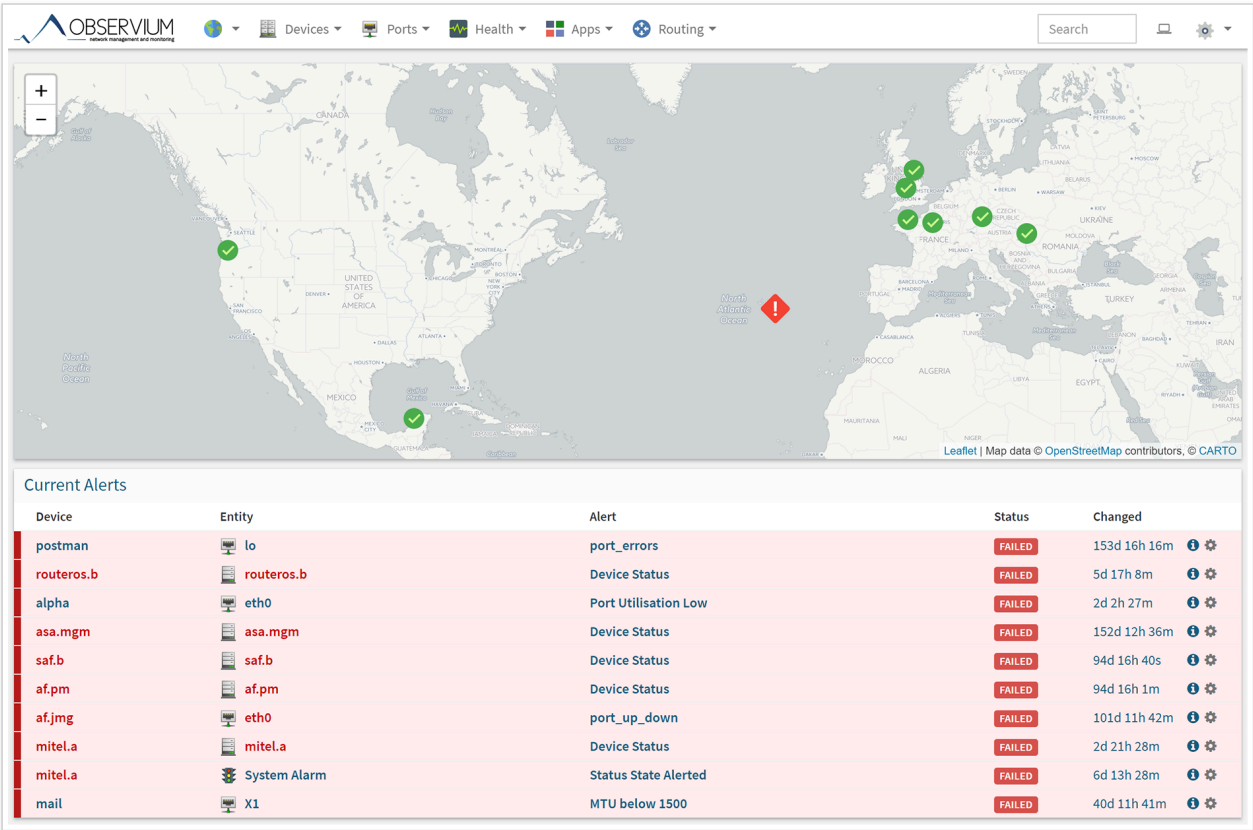
Observium also has built-in security features such as support for SNMPv3, which provides enhanced security features such as authentication and encryption for SNMP communication. Observium also supports SNMP traps for receiving notifications from devices in a secure manner.

In addition, Observium provides options for data retention and archiving, allowing for long-term storage of monitoring data for compliance and historical analysis purposes. This helps ensure data integrity and availability even over extended periods of time.

Observium also actively maintains and releases regular updates and security patches to address known vulnerabilities and ensure the security of the system. In

summary, Observium is designed to be scalable and secure, providing efficient and reliable monitoring of large networks while ensuring the confidentiality, integrity, and availability of monitoring data.

Front Page



5.1 Front Page

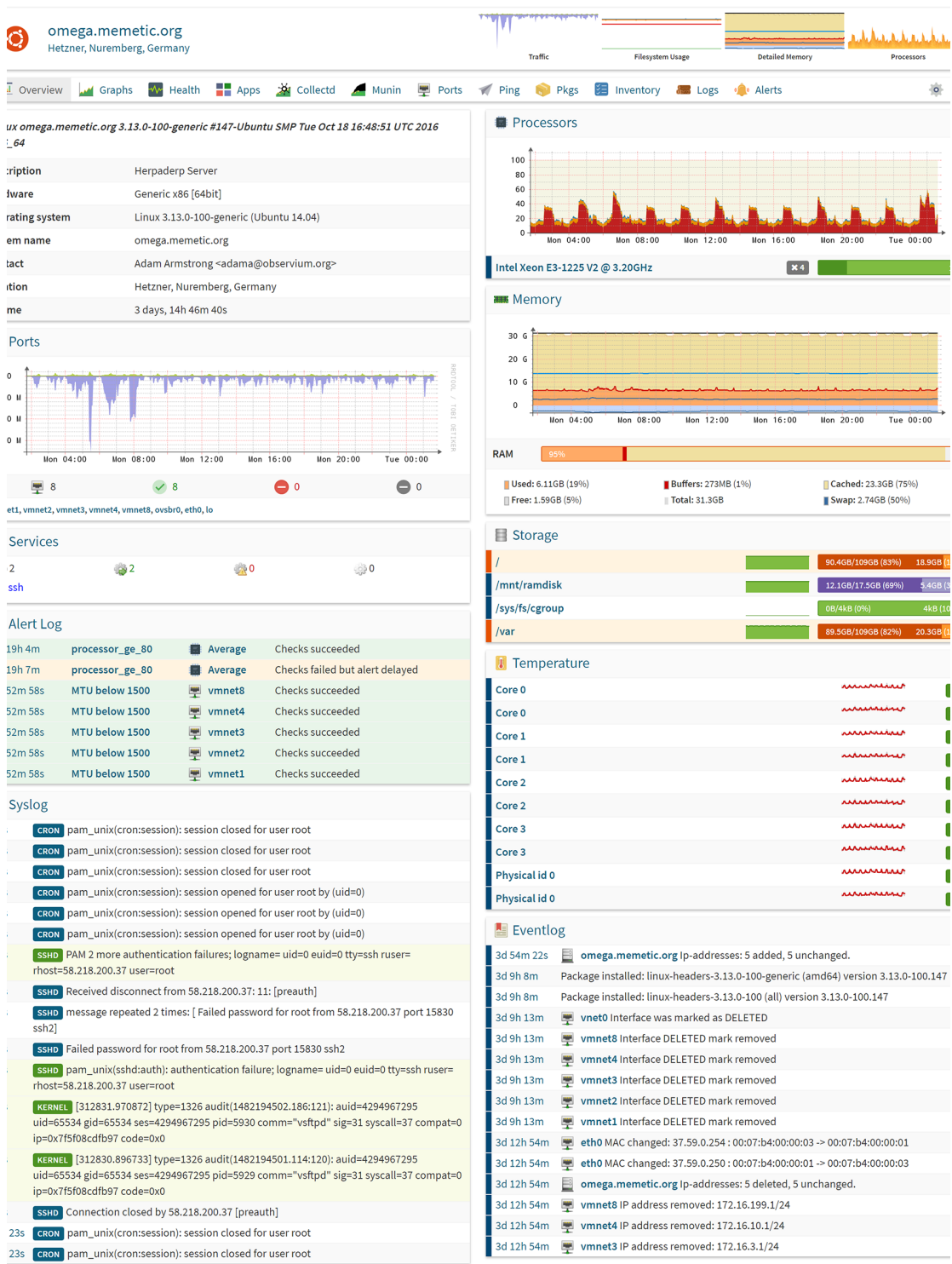


Fig 5.2 Linux Details



Fig 5.3 Port Details in Observium

Alerting

Alerts

Alert Checkers

Alert Logging

Scheduled Maintenance

Syslog Alerts

Syslog Rules

Contacts

Rebuild

Add Syslog Rule

Add Checker

Devices

Checkers

Status Type

Date presets

From

To

Search

26.8k Items

123456789268269Next

100

Date	Device	Alert Check	Entity	Message	Status	Notified
2016-12-20 00:45:05	gcom.s3650	Device Status	gcom.s3650	Checks failed but alert delayed	FAIL_DELAYED	NO
2016-12-19 23:29:16	gcom.s3650	Device Status	gcom.s3650	Checks succeeded	OK	NO
2016-12-19 23:20:09	gcom.s3650	Device Status	gcom.s3650	Checks failed but alert delayed	FAIL_DELAYED	NO
2016-12-19 23:09:23	asa.mgm	Device Status	asa.mgm	Alert notification sent	ALERT_NOTIFY	NO
2016-12-19 22:39:23	postman	port_errors	lo	Alert notification sent	ALERT_NOTIFY	YES
2016-12-19 22:38:55	alpha	Port Utilisation Low	eth0	Alert notification sent	ALERT_NOTIFY	YES
2016-12-19 22:38:04	af.jmg	port_up_down	eth0	Alert notification sent	ALERT_NOTIFY	YES
2016-12-19 20:24:10	lancom	Device Status	lancom.l310agn	Checks succeeded	OK	NO
2016-12-19 20:18:20	lancom	Device Status	lancom.l310agn	Checks failed but alert delayed	FAIL_DELAYED	NO
2016-12-19 18:28:35	routeros.a	Device Status	routeros.a	Checks succeeded	OK	NO
2016-12-19 18:23:11	routeros.a	Device Status	routeros.a	Checks failed but alert delayed	FAIL_DELAYED	NO
2016-12-19 15:04:19	mail	MTU below 1500	X1	Alert notification sent	ALERT_NOTIFY	YES
2016-12-18 22:37:46	af.jmg	port_up_down	eth0	Alert notification sent	ALERT_NOTIFY	YES
2016-12-18 22:34:27	alpha	Port Utilisation Low	eth0	Alert notification sent	ALERT_NOTIFY	YES
2016-12-18 16:43:34	beta	Device Status	beta.memetic.org	Checks succeeded	OK	NO
2016-12-18 16:38:29	beta	Device Status	beta.memetic.org	Checks failed but alert delayed	FAIL_DELAYED	NO
2016-12-18 14:59:33	mail	MTU below 1500	X1	Alert notification sent	ALERT_NOTIFY	YES
2016-12-18 13:34:38	saf.a	Device Status	saf.a	Alert notification sent	ALERT_NOTIFY	NO
2016-12-18 13:34:37	saf.b	Device Status	saf.b	Alert notification sent	ALERT_NOTIFY	NO
2016-12-18 13:33:36	af.pm	Device Status	af.pm	Alert notification sent	ALERT_NOTIFY	NO
2016-12-18 13:03:11	af.jmg	Device Status	af.jmg.herpaderp.domain.com	Alert notification sent	ALERT_NOTIFY	NO
2016-12-18 11:53:11	mitel.a	Status State Alerted	System Alarm	Alert notification sent	ALERT_NOTIFY	YES
2016-12-18 11:24:07	sapito	processor_ge_80	Average	Checks succeeded	OK	NO
2016-12-18 11:19:05	sapito	processor_ge_80	Average	Checks failed but alert delayed	FAIL_DELAYED	NO

26.8k Items

123456789268269Next

100

Fig 5.4 Alert Logging

Alerting	Alerts	Alert Checkers	Alert Logging	Scheduled Maintenance	Syslog Alerts	Syslog Rules	Contacts	Rebuild	Add Syslog Rule	Add Checker
Alert Types	All	Port	Memory	Processor	Storage	BGP Peer	Device	Status	Filter (All)	
183 Items	1	2	Next							# 100
Device	Entity	Alert	Status	Checked	Changed	Alerted				
c1800-a	Fan 1	Status State Alerted	OK	40s	339d 5h 10m	Never				
c1800-b	Fan 1	Status State Alerted	OK	47s	339d 5h 10m	Never				
apc-b	Diagnostics Results (last 2016-12-15)	Status State Alerted	OK	45s	89d 12h 31m	Never				
at8000s	PoE Group 4	Status State Alerted	OK	4m 47s	68d 49m 48s	Never				
at8000s	PoE Group 3	Status State Alerted	OK	4m 47s	68d 49m 48s	Never				
at8000s	PoE Group 2	Status State Alerted	OK	4m 47s	68d 49m 48s	Never				
at8000s	PoE Group 1	Status State Alerted	OK	4m 47s	68d 49m 48s	Never				
asr1k	Cisco ASR1001 AC Power Supply	Status State Alerted	OK	4m 46s	199d 39m 47s	Never				
asr1k	Cisco ASR1001 AC Power Supply	Status State Alerted	OK	4m 47s	199d 39m 47s	Never				
drac	Overall System Status	Status State Alerted	OK	29s	339d 5h 9m	Never				
netcaler	HA Mode	Status State Alerted	OK	4m 42s	225d 49m 40s	Never				
netcaler	HA State	Status State Alerted	OK	4m 43s	276d 17h 59m	Never				
cisco	module 8 power-output-fail	Status State Alerted	OK	4m 43s	283d 11h 49m	Never				
cisco	module 5 power-output-fail	Status State Alerted	OK	4m 43s	283d 11h 49m	Never				
cisco	module 4 power-output-fail	Status State Alerted	OK	4m 43s	283d 11h 49m	Never				
cisco	Chassis Fan Tray 2	Status State Alerted	OK	4m 43s	339d 5h 9m	Never				
cisco	Chassis Fan Tray 1	Status State Alerted	OK	4m 43s	339d 5h 9m	Never				
mitel.a	System Alarm	Status State Alerted	FAILED	2d 21h 41m	6d 13h 21m	12h 56m 47s				
catos-6509	1300 watt supply AC	Status State Alerted	OK	3m 31s	199d 39m 42s	Never				
catos-6509	1300 watt supply AC	Status State Alerted	OK	3m 32s	199d 39m 42s	Never				
catos-6509	VTT - env temp	Status State Alerted	OK	3m 32s	339d 5h 9m	Never				
183 Items	1	2	Next							# 100

Fig 5.5 Alerts