**Internship Project**

**Of**



A

PROJECT REPORT

ON **“360 Degree Monitoring”**

BY

**Tribe - O**

**SUBMITTED BY**

|  |  |
| --- | --- |
| OMKAR SAMBHAJI DAMAME | ABHAY SINGH BAJETA (LEADER) |
| PRASAD MOHAN DHUPKAR (CO-LEADER) | SUNITA KUMARI NAYAK |
| DEEPAK PANIGRAHI | SANTOSINI SATAPATHY |
| SUBHANGI SUBHADARSHINI NAYAK | SANTOSHINI NAIK |
| SIDDIK JAHANGIR ATTAR | ABHIPSA SAMANTRA |
|  | BIDYUT RANJAN NAYAK |

**UNDER THE GUIDANCE OF**

Mr. Dinesh Behera

**ABSTRACT**

When selecting an infrastructure-monitoring suite, you need to take into account several factors to find the perfect match for your needs. First, select a tool on the basis of the required functionalities so that it aligns with your technical and business needs. Next, evaluate the deployment and maintenance factors to match the tool with your IT team's resources and capabilities. Finally, with a proper understanding of how the tool will affect your organization, calculate the total cost of ownership. Many company use monitoring technologies in order to monitor their infrastructure such as CloudWatch of AWS, Cloud Monitoring of Google Cloud, Azure Monitor of Microsoft Azure which just work fine but they costs a lot, can be only used on their respective platforms, not enough customization of dashboard, need to implement custom metrics which can be a very tedious task. Also it is very expensive at the enterprise level - can be over $50,000 per year. So, we decied to develop a single solution which can do various kinds of monitorings such as application/webserver monitoring, database monitoring, network monitoring, server monitorong, virtualization monitoring, etc. This solution just require few instances and there are no charges for how API calls we do or how many time information we pull from clients which would we monitor. Also the tools are used for internal use only so that only employees who are granted permission, can use the tool. Our solution gives nearly all kinds of custom monitoring so the name is 360 Degree monitoring.

**Contents**

1. Introduction
2. Approach to Proposed System
   1. Existing System
   2. Need for New System
3. Feasibility Study
   1. Technical Feasibility
   2. Economic Feasibility
   3. Operational Feasibility
4. Operating Environments
   1. Hardware Requirement
   2. Software Requirement
5. Technologies Used
6. Architecture
   1. Nagios + NRPE with Plugins
   2. Zabbix Architecture
7. Output details
8. Conclusion

**1. Introduction**

The project, “360 Degree Monitoring” is a suite of technologies used for server monitoring, network monitoring, database monitoring, webserver monitoring and virtualization monitoring. It helps system admin to monitor the company’s infrastructure extensively and prevent any hazards even before occurring due to end-to-end monitoring. Having a 360-degree monitoring not only efficiently prevents malfunction of any part of the infra or software but provides the user tracking.

The respective system admins and others who are listed as management of infra get notified whenever there are any issues with the infra and webservers as soon as possible using email notifications.

The 360 Degree Monitoring is used by two roles,

1. Admin - The Admin role is used by admins for
   * 1. Management of tools
     2. Report generation
     3. Raise tickets to report issues.
2. Engineers - The Engineer role is support engineers for
   * 1. Respond to ticket generated by admins.
     2. Resolution of issues.

Also, the monitoring tools are accessible via both mobile and desktops due to their responsive design for any browser.

For the development of this project, we used new technologies which are – Nagios-Core, Zabbix, Postfix, AWS Infrastructure, VMware and so on.

**2. Approach to Proposed System**

*2.1 Existing Systems*

The company uses monitoring technologies in order to monitor their infrastructure such as CloudWatch of AWS, Cloud Monitoring of Google Cloud, Azure Monitor of Microsoft Azure which just work fine but they cost a lot, can be only used on their respective platforms, not enough customization of dashboard, need to implement custom metrics which can be a very tedious task. They are platform dependent and also it is very expensive at the enterprise level - can be over $50,000 per year. Also, there is very limited customization options present.

*2.2 Need for New System*

We decide to develop a single solution which can do various kinds of monitoring such as application/webserver monitoring, database monitoring, network monitoring, server monitoring, virtualization monitoring, etc. This solution just requires few instances and there are no charges for how API calls we do or how many time information we pull from clients which would we monitor. Also, the tools are used for internal use only so that only employees who are granted permission, can use the tool. With automation, integrating cost control into your daily processes can be straightforward.

Our solution gives nearly all kinds of custom monitoring so the name of the project is 360 Degree monitoring.

**3. Feasibility study**

Feasibility study is to check the viability of the project under consideration. Theoretically various types of feasibilities can be conducted, here we have conducted three types of feasibilities explained as under.

*3.1 Technical Feasibility*

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at this point of time, there are multiple detailed designs of the system, making it easy to access issues like performance, costs on (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis. Since the technology we are using is easily available for the organization so the system is considered to be technical feasible.

*3.2 Economic Feasibility*

The new system is mobile application so evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at this point of time, there are few detailed designs of the system out market, making it somewhat easy to access issues like performance, costs on (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis. Since the technology we are using is easily available for the organization so the system is considered to be technical feasible. That it can be used by any person. And in future it may open source it depends on company policy. So that for using this it is feasible for anyone. And the cost of system is bearable by company.

*3.3 Operational Feasibility*

This system helps us to remove the limitation in existing system and some more new feature will be added to this application, so that user can have great look, features and improved user-friendly. User can have new features like log generation on demand, custom metrics, added dashboards, no limit on API calls and how many metrics you pull number of times. Also, the system can be ported for another infrastructure with few modifications.

**4.** **Operation Environment - Hardware and Software**

*4.1 Hardware Requirement*

|  |  |  |
| --- | --- | --- |
| **Components** | **Minimum** | **Recommended** |
| Processor | 1 Core 1 GHz | 2 GHz |
| RAM | 1 GB | 2 GB |
| Storage | 8 GB HDD | 40 GB HDD |
| Connection | 512 Kbps | 1 Mbps |
| Instance | t2.micro (AWS) or similar | t3a.small (AWS) or similar |

*4.2 Software Requirement*

|  |  |  |
| --- | --- | --- |
| **Components** | **Minimum** | **Recommended** |
| OS | Ubuntu 16.04 (CLI) | Ubuntu 20.04 |
| Nagios - Core | 4.4.6 | 4.4.7 |
| Zabbix | 6.0.0 | 6.0.3 |
| Postfix | 3.4.5 | 3.7.2 |
| Email | 1 Email Account | 2 Email Accounts (Gmail Preferred) |

**5. Technologies Used**

Basically, the project is developed in the dynamic environment so it has used many new technologies. Like Nagios, Zabbix, Postfix, AWS and many more. The description of all these is below.

* 1. *Nagios-Core:*

Nagios-core is a cross-platform monitoring tool that runs on Windows, Linux/Unix, and Mac OS/X machines. Its features include both active and passive checks, remote management, and a local monitoring interface.

NRPE (Nagios Remote Plugin Executor) is an addon allows you to remotely execute Nagios plugins on remote Linux/Unix machines. This allows you to monitor remote machine metrics (disk usage, CPU load, etc.). NRPE can also communicate with some Windows agent addons like NSClient++, so you can execute scripts and check metrics on remote Windows machines as well.

Plugins are also known as NRPE Addons which collects the respective information from the server which then passed along to NRPE of the server which is then interpreted by Nagios-Core and displayed accordingly on the dashboard. In order to monitor the NRPE and plugins are mandatorily installed on the remote hosts.

* 1. *Zabbix:*

Zabbix is an open-source software tool to monitor IT infrastructure such as networks, servers, virtual machines, and cloud services. Zabbix collects and displays various metrics. Keep control of your infrastructure by collecting any metric from any source. Zabbix provides its users with very flexible, intelligent threshold definition options. While a threshold for trigger may be as simple as "bigger than x", it is possible to use all power of supported functions and operators for statistical analysis of history data.

Uses multiple messaging channels to notify the responsible person or people about the different kinds of events occurring in your environment. Custom alert messages can be defined as different messages for different messaging channels. You can either utilize the default message templates or create and customize your own message template.

One can Gain additional insights and extend observability by powerful data visualization such as graphs, geo-maps, infrastructure maps and can also schedule report and downtimes.

A native Zabbix agent, developed in C language, may run on various supported platforms, including Linux, UNIX and Windows, and collect data such as CPU, memory, disk and network interface usage from a device. Due to its small footprint, the agent can be run on devices with limited resources. The monitoring configs are centralized in Zabbix, making it easier to manage the Zabbix agent, which can use a single configuration file on all servers.

* 1. *Postfix:*

Postfix is a free and open-source mail transfer agent (MTA) that routes and delivers electronic mail. It is Wietse Venema's mail server that started life at IBM research as an alternative to the widely-used Sendmail program. Now at Google, Wietse continues to support Postfix.

As an SMTP server, Postfix implements a first layer of defense against spambots and malware. Administrators can combine Postfix with other software that provides spam/virus filtering, message-store access, or complex SMTP-level access-policies. Postfix implements a high-performance parallelized mail-delivery engine. Postfix is often combined with mailing-list software (such as Mailman or Sendmail).

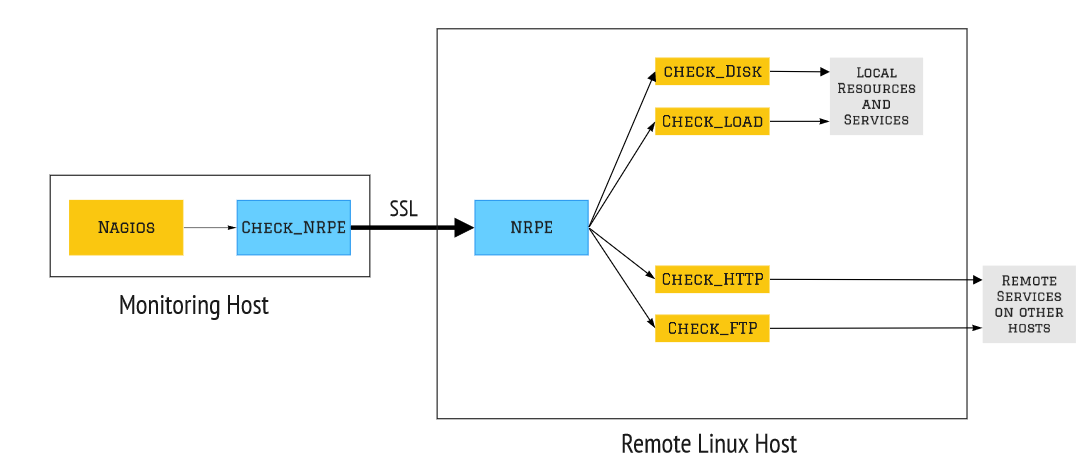
* 1. *AWS:*

AWS, also known as Amazon Web Services, a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis. One of these services is Amazon Elastic Compute Cloud (EC2), which allows users to have at their disposal a virtual cluster of computers, available all the time, through the Internet.

AWS services are delivered to customers via a network of AWS server farms located throughout the world. Fees are based on a combination of features chosen by the subscriber required availability, redundancy, security, and service options. Subscribers can pay for a single virtual AWS computer, a dedicated physical computer, or clusters of either.

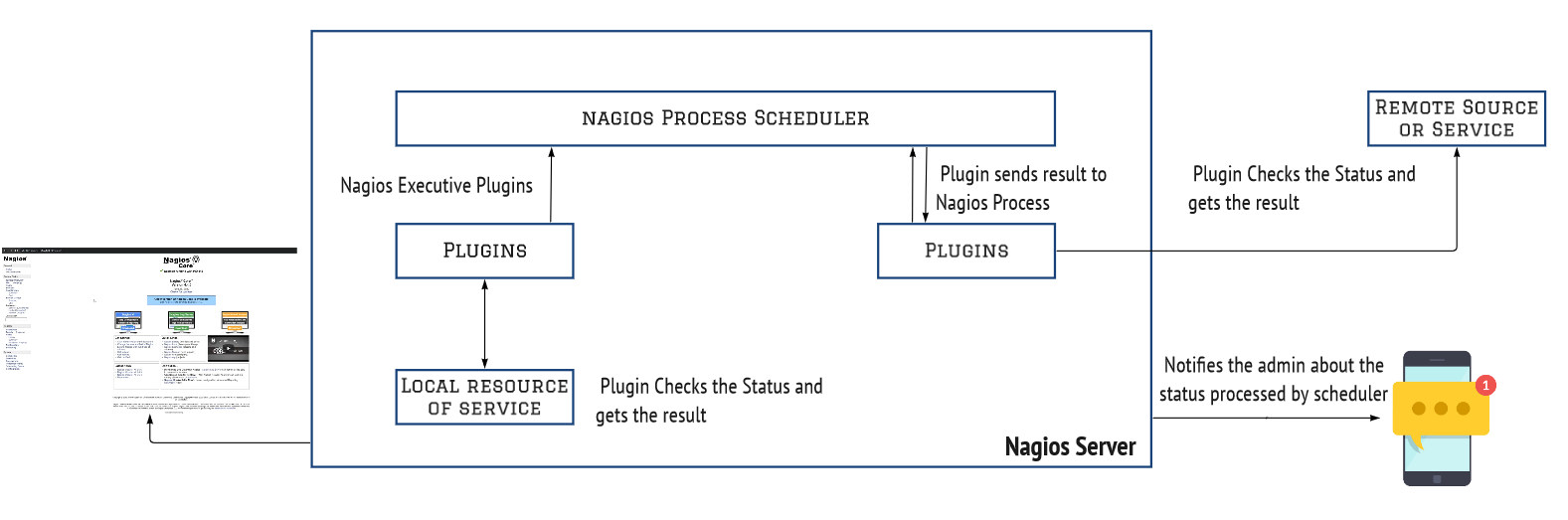
**6. Architecture**

*6.1 Nagios + NRPE with Plugins*



In this above image we are showing a simpler layout on how this remote monitoring of a Linux server. In which as you can see, we use the check\_nrpe command to make it communicate with the remote host’s NRPE module using and SSL connection which then uses the installed plugins on the remote server and collects all the information.

Also, to get to know what really happens in deep is shown in the architecture of Nagios which is as follows:



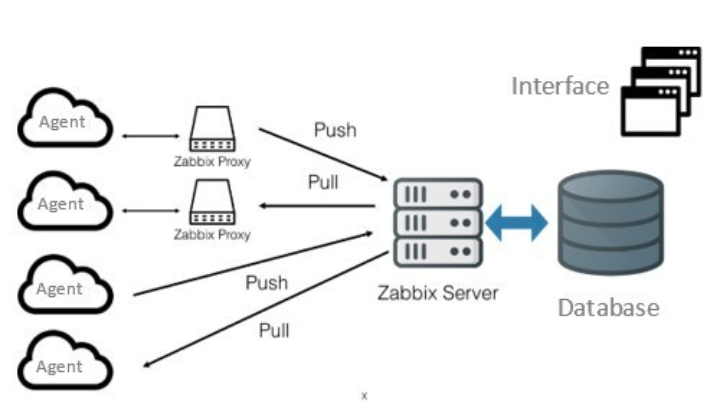
Nagios Process Scheduler is the main entity that makes this all happen, but to let you know that first we need to add all the plugins required for this monitoring on the remote host itself without those plugins Nagios will not be able to get any information it needs.

Now in the architecture as you can see the Nagios Process Scheduler communicates first with the Nagios plugins in remote monitoring as well as to monitor itself locally.

Plugins like check\_disk which checks the disk status, check\_http which checks the status of that port on which you might have hosted a website or a webserver etc.

These plugins are then used to collect information and using NRPE they send it back to the main process scheduler. This system is used in the same way to monitor local resources as well as to monitor any server remotely. In case of any problems, it notifies the admin of that server using SMS or E-Mail to resolve it. All of these things are shown to us using the web interface that is hosted on the internet.

*6.2 Zabbix Architecture*

Zabbix consists of several major software components. Their responsibilities are outlined below.

**Server**

Zabbix server is the central component to which agents report availability and integrity information and statistics. The server is the central repository in which all configuration, statistical and operational data are stored.

**Database storage**

All configuration information as well as the data gathered by Zabbix is stored in a database.

**Web interface**

For an easy access to Zabbix from anywhere and from any platform, the web-based interface is provided. The interface is part of Zabbix server, and usually (but not necessarily) runs on the same physical machine as the one running the server.

**Proxy**

Zabbix proxy can collect performance and availability data on behalf of Zabbix server. A proxy is an optional part of Zabbix deployment; however, it may be very beneficial to distribute the load of a single Zabbix server.

**Agent**

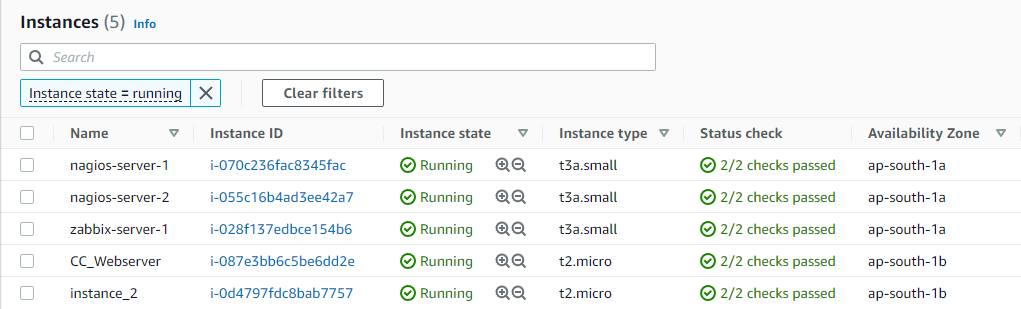
Zabbix agents are deployed on monitoring targets to actively monitor local resources and applications and report the gathered data to Zabbix server. Since Zabbix 4.4, there are two types of agents available: Zabbix agent (lightweight, supported on many platforms, written in C) and Zabbix agent 2 (extra-flexible, easily extendable with plugins, written in Go).

**Data flow**

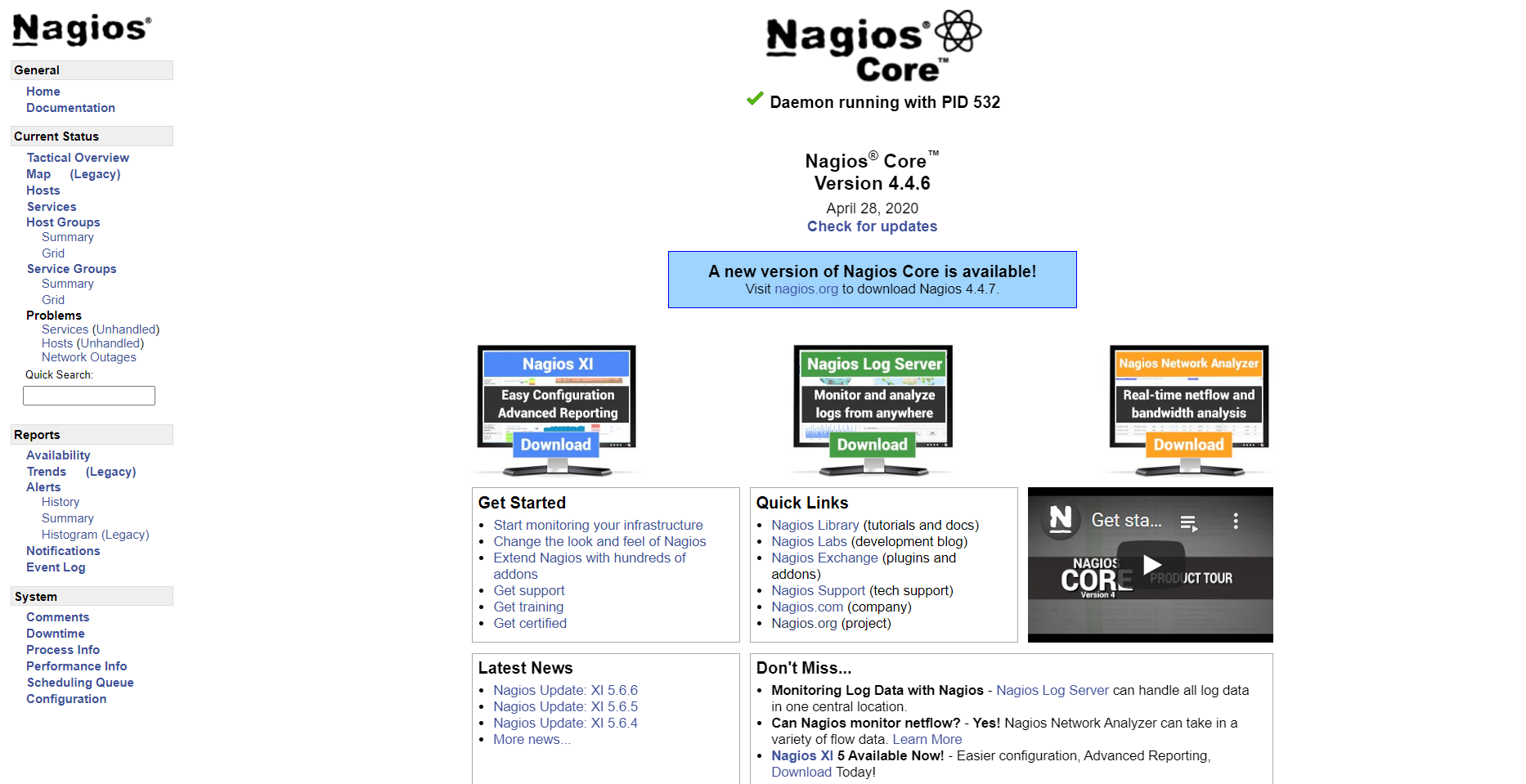
In addition, it is important to take a step back and have a look at the overall data flow within Zabbix. In order to create an item that gathers data you must first create a host. Moving to the other end of the Zabbix spectrum you must first have an item to create a trigger. You must have a trigger to create an action. Thus, if you want to receive an alert that your CPU load is too high on *Server X* you must first create a host entry for *Server X* followed by an item for monitoring its CPU, then a trigger which activates if the CPU is too high, followed by an action which sends you an email. While that may seem like a lot of steps, with the use of templating it really isn't. However, due to this design it is possible to create a very flexible setup.

**7. Output details**

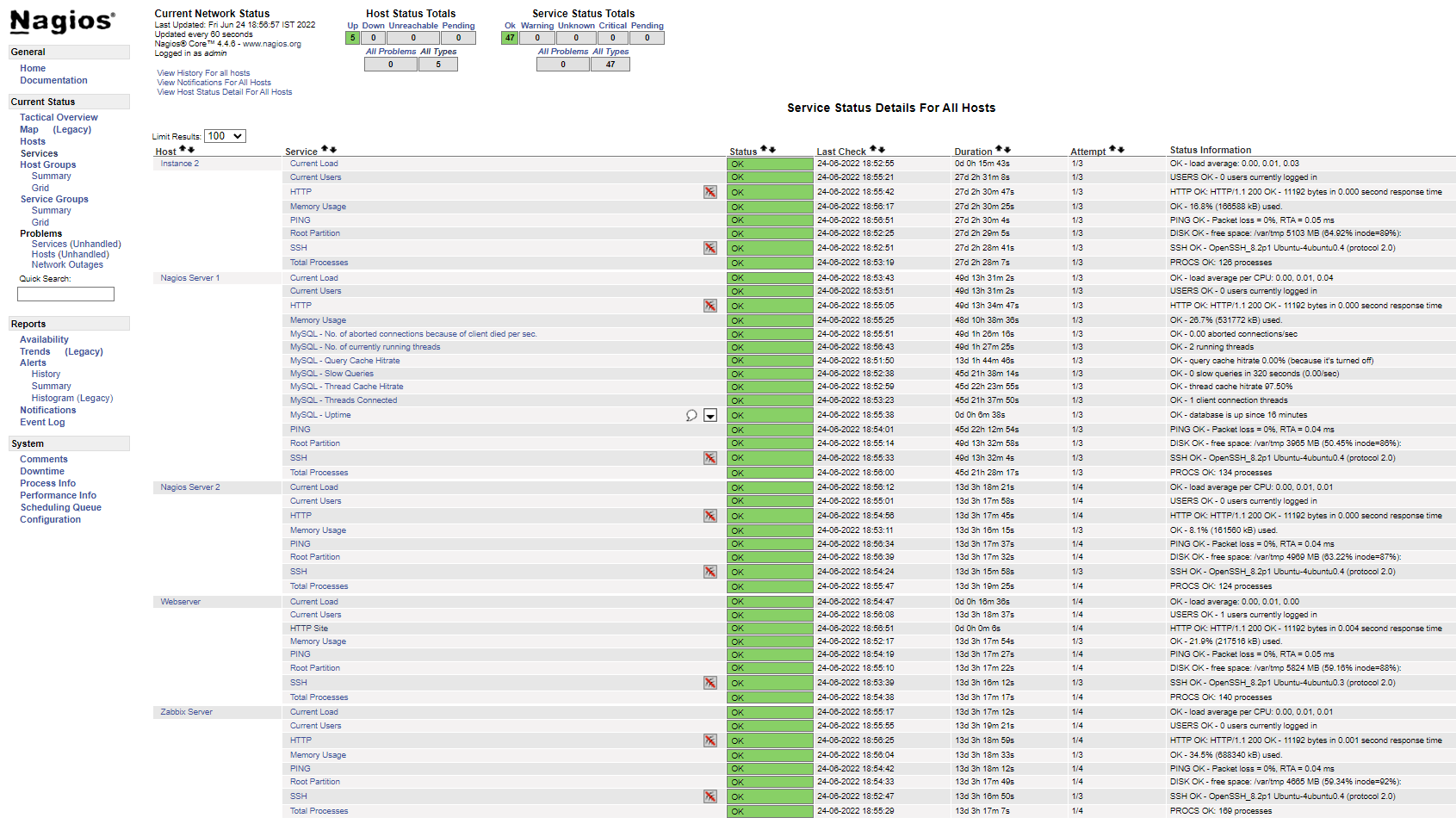
Nagios + Zabbix server with clients running on AWS infrastructure



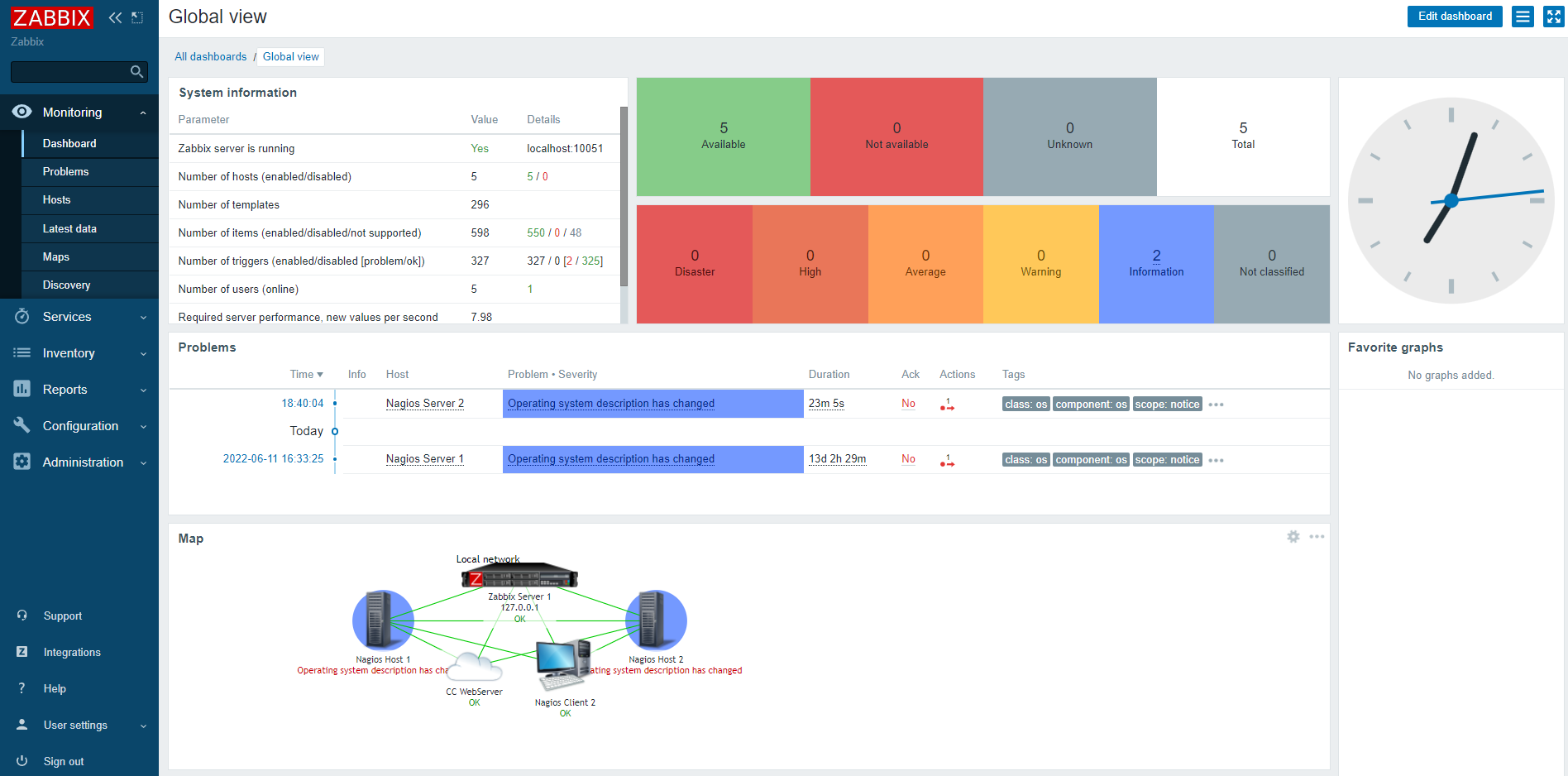
Nagios Web Interface



All processes and their status being monitored by Nagios Server



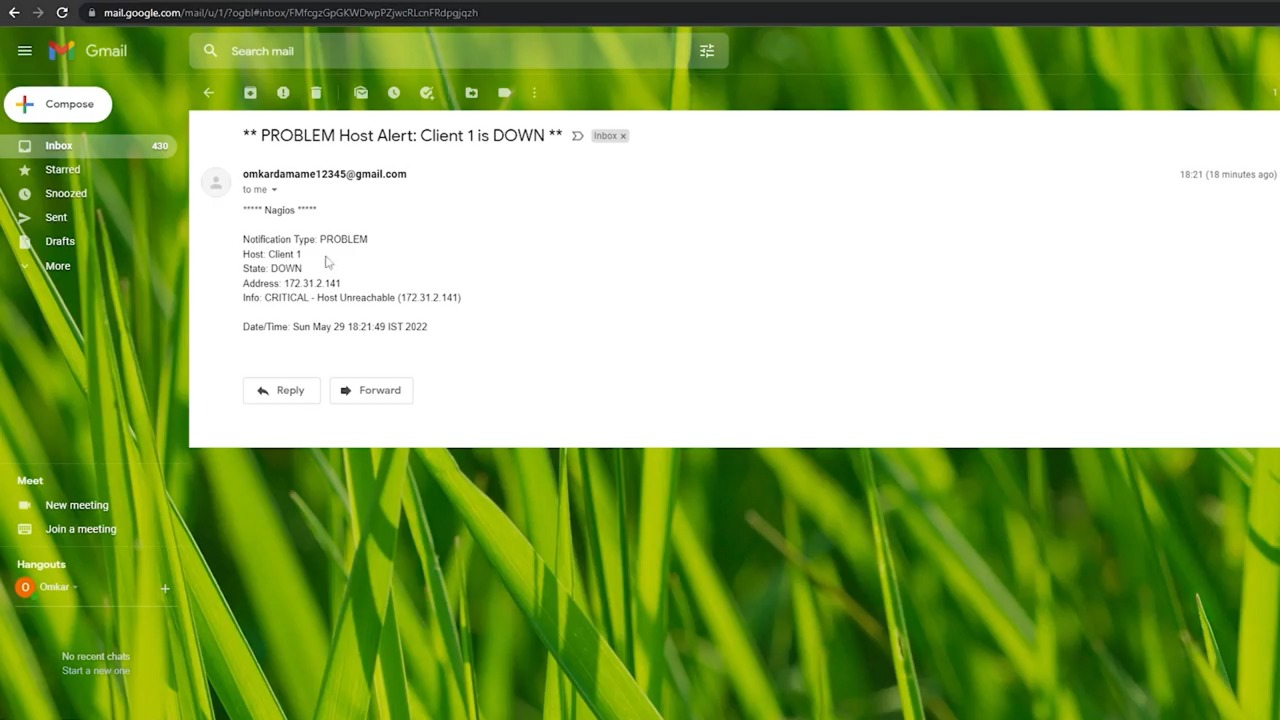
Zabbix Web Interface



Monitoring by Graphs in Zabbix



Email notification alert by Nagios/Zabbix



**8. CONCLUSION**

After all of these efforts we are finally able to conclude this report by saying that we have achieved the main objective of our project that is monitoring multiple instances of AWS using open-source tools. This solution to monitoring is very cost effective than other tools like AWS CloudWatch. Also, to add to the point of cost effectiveness, this solution to monitoring was free but wasn’t easy to deploy, but to overcome this problem we have used automated Bash scripts created by us that let these tools installed automatically on Ubuntu 20.04 OS. Also, user don’t have to worry about configuring these tools by himself because we have added the configuration of these tools in those automated Bash scripts as well.