Internship Project

Of

PRODEVANS

A

PROJECT REPORT

ON

"360 Degree Monitoring"

BY

Tribe - O

SUBMITTED BY

OMKAR SAMBHAJI DAMAME

PRASAD MOHAN DHUPKAR (CO-LEADER)

DEEPAK PANIGRAHI

SUBHANGI SUBHADARSHINI NAYAK

SIDDIK JAHANGIR ATTAR

ABHAY SINGH BAJETA (LEADER)

SUNITA KUMARI NAYAK

SANTOSINI SATAPATHY

SANTOSHINI NAIK

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

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
    - i) Management of tools
    - ii) Report generation
    - iii) Raise tickets to report issues.
2) Engineers - The Engineer role is support engineers for
    - i) Respond to ticket generated by admins.
    - ii) 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.

5.1 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.

5.2 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.

5.3 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).

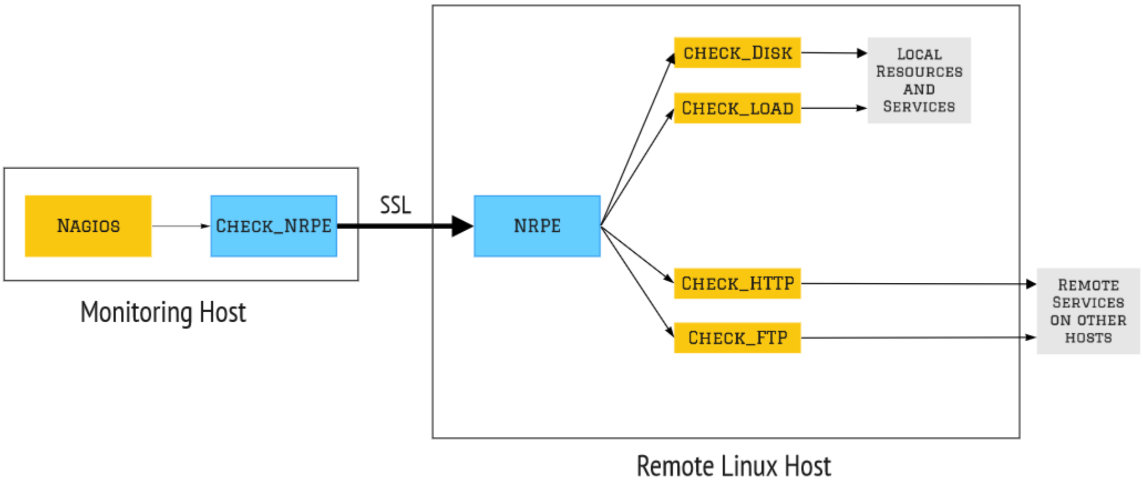
5.4 AWS:

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.

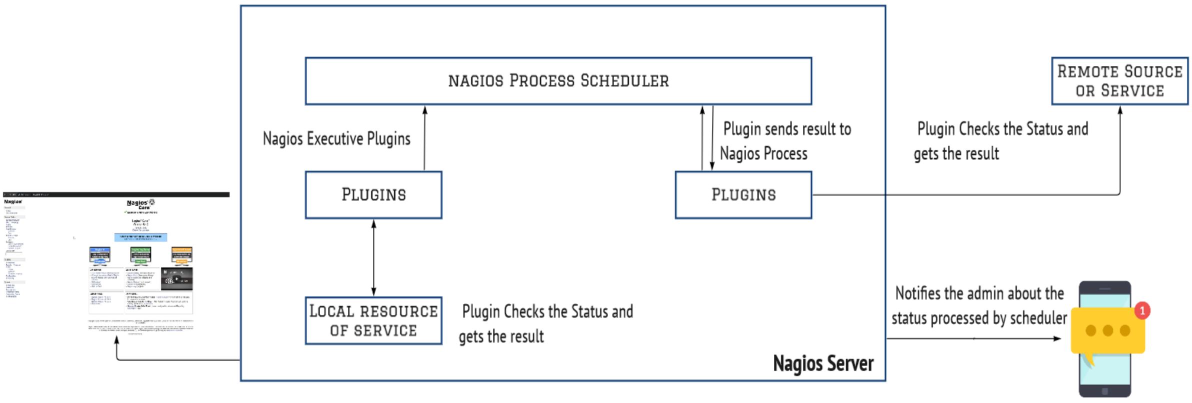
6. Architecture

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
Nagios Executive Plugins
Plugin sends result to Nagios Process
PLUGINS
PLUGINS
REMOTE SOURCE OR SERVICE
Plugin Checks the Status and gets the result
LOCAL RESOURCE OF SERVICE
Plugin Checks the Status and gets the result
Notifies the admin about the status processed by scheduler
Nagios Server

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.

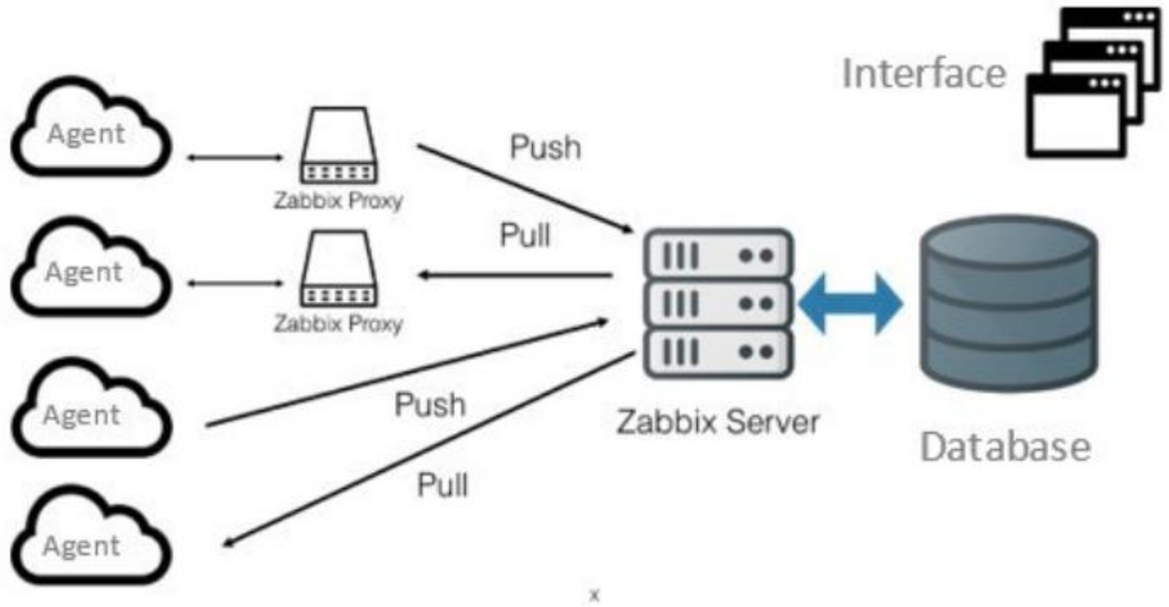
Now in the architecture as you can see the Nagios Process Scheduler communicates first with the Nagios plugins in remote monitoring as well as to monitor itself locally.

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

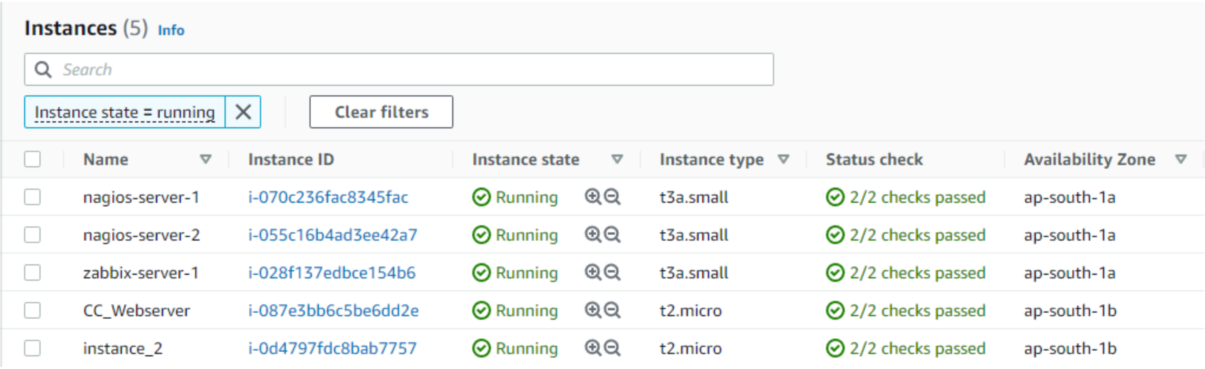
Zabbix agents are deployed on monitoring targets to actively monitor local resources and applications and report the gathered data to Zabbix server. Since Zabbix 4.4, there are two types of agents available: Zabbix agent (lightweight, supported on many platforms, written in C) and Zabbix agent 2 (extra-flexible,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



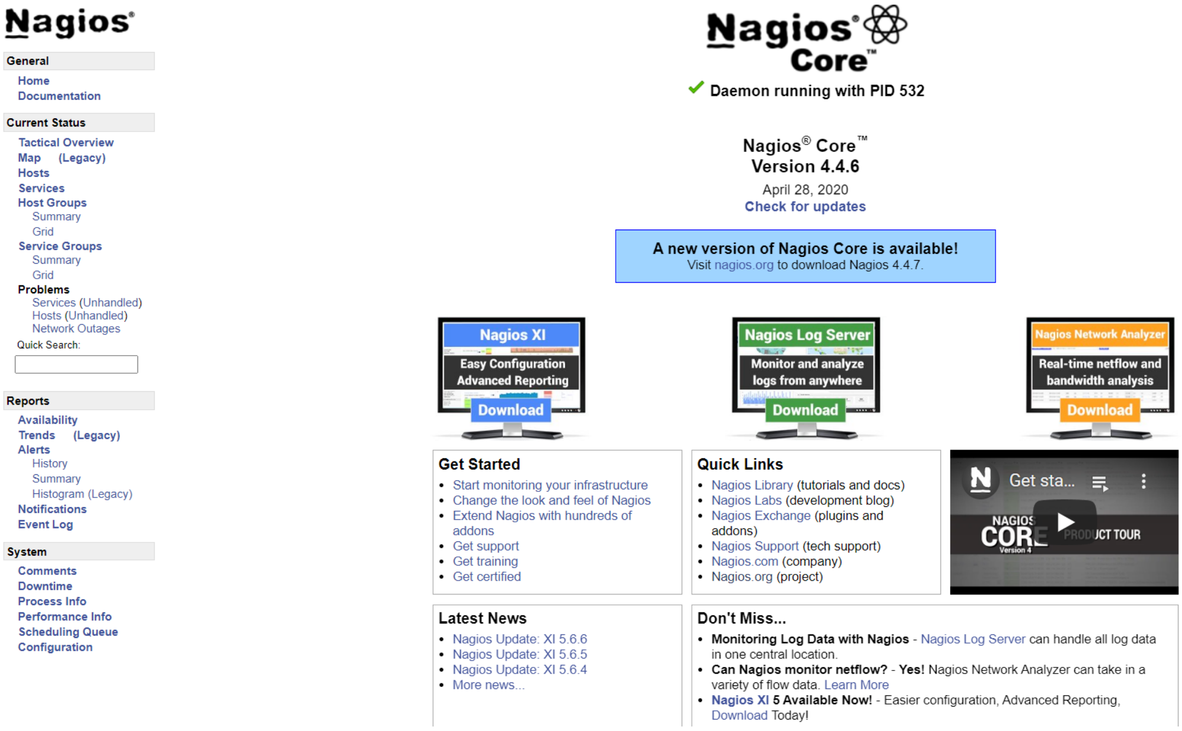
Instances (5) Info

Search

Instance state = running ✕ | Clear filters

□	Name	Instance ID	Instance state	Instance type	Status check	Availability Zone
□	nagios-server-1	i-070c236fac8345fac	✓ Running	t3a.small	✓ 2/2 checks passed	ap-south-1a
□	nagios-server-2	i-055c16b4ad3ee42a7	✓ Running	t3a.small	✓ 2/2 checks passed	ap-south-1a
□	zabbix-server-1	i-028f137edbce154b6	✓ Running	t3a.small	✓ 2/2 checks passed	ap-south-1a
□	CC_Webserver	i-087e3bb6c5be6dd2e	✓ Running	t2.micro	✓ 2/2 checks passed	ap-south-1b
□	instance_2	i-0d4797fdc8bab7757	✓ Running	t2.micro	✓ 2/2 checks passed	ap-south-1b

Nagios Web Interface



Nagios®

General

- Home
- Documentation

Current Status

- Tactical Overview
- Map (Legacy)
- Hosts
- Services
- Host Groups
  - Summary
  - Grid
- Service Groups
  - Summary
  - Grid
- Problems
  - Services (Unhandled)
  - Hosts (Unhandled)
  - Network Outages

Quick Search:

Reports

- Availability
- Trends (Legacy)
- Alerts
  - History
  - Summary
  - Histogram (Legacy)
- Notifications
- Event Log

System

- Comments
- Downtime
- Process Info
- Performance Info
- Scheduling Queue
- Configuration

Nagios® Core™

✓ Daemon running with PID 532

Nagios® Core™
Version 4.4.6
April 28, 2020
Check for updates

A new version of Nagios Core is available!
Visit nagios.org to download Nagios 4.4.7.

Get Started

- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of addons
- Get support
- Get training
- Get certified

Quick Links

- Nagios Library (tutorials and docs)
- Nagios Labs (development blog)
- Nagios Exchange (plugins and addons)
- Nagios Support (tech support)
- Nagios.com (company)
- Nagios.org (project)

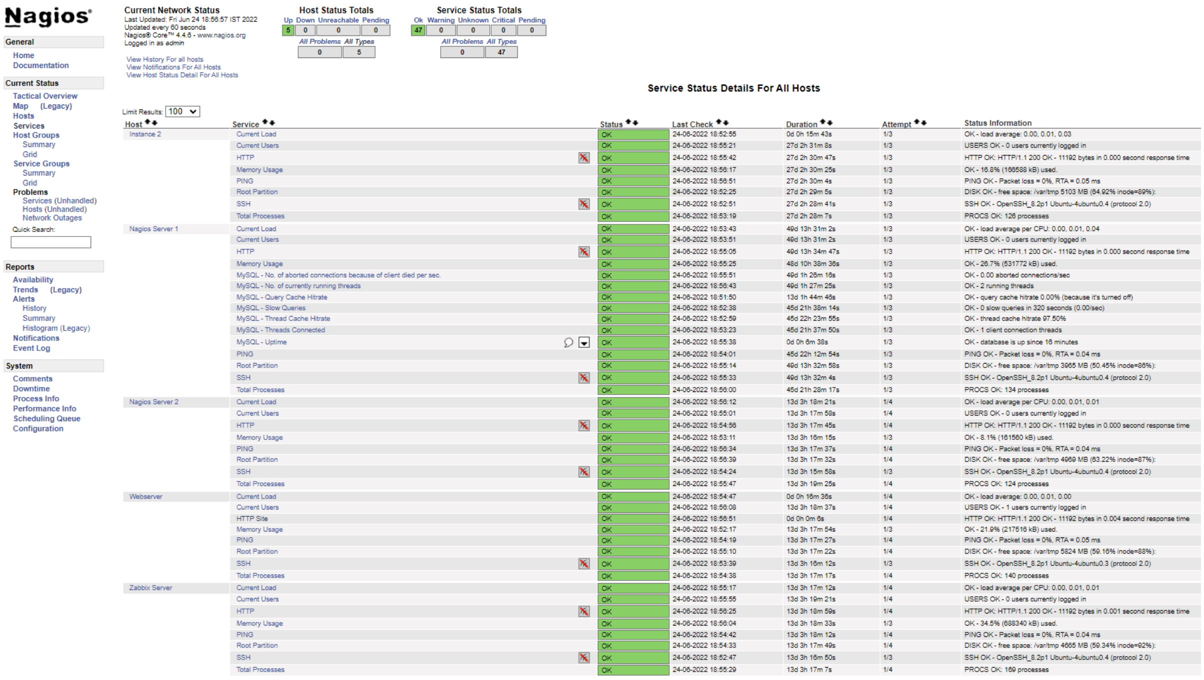
Latest News

- Nagios Update: XI 5.6.6
- Nagios Update: XI 5.6.5
- Nagios Update: XI 5.6.4
- More news...

Don't Miss...

- **Monitoring Log Data with Nagios** - Nagios Log Server can handle all log data in one central location.
- **Can Nagios monitor netflow?** - **Yes!** Nagios Network Analyzer can take in a variety of flow data. Learn More
- **Nagios XI 5 Available Now!** - Easier configuration, Advanced Reporting, Download Today!

All processes and their status being monitored by Nagios Server



Current Network Status
Last Updated: Fri Jun 24 18:56:57 IST 2022
Updated every 60 seconds
Nagios® Core™ 4.4.6 - www.nagios.org
Logged in as *admin*

View History For all hosts
View Notifications For All Hosts
View Host Status Detail For All Hosts

Host Status Totals

Up	Down	Unreachable	Pending
5	0	0	0

All Problems	*All Types*
0	5

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
47	0	0	0	0

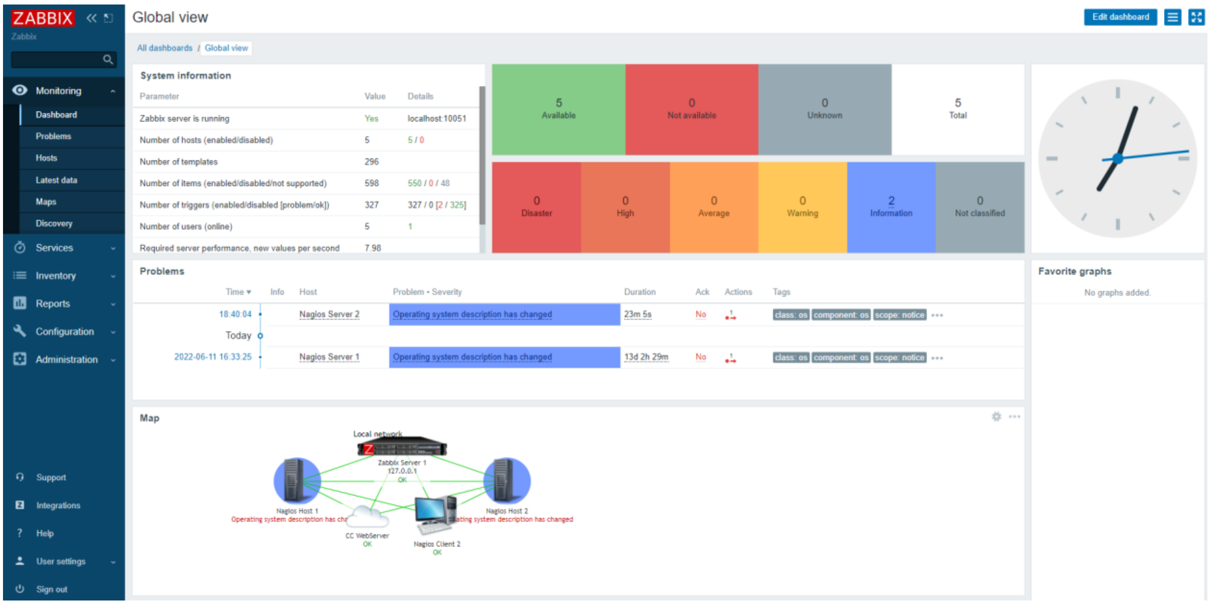
All Problems	*All Types*
0	47

Service Status Details For All Hosts

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
Instance 2	Current Load	OK	24-06-2022 18:52:55	0d 0h 15m 43s	1/3	OK - load average: 0.00, 0.01, 0.03
	Current Users	OK	24-06-2022 18:55:21	27d 2h 31m 8s	1/3	USERS OK - 0 users currently logged in
	HTTP	OK	24-06-2022 18:55:42	27d 2h 30m 47s	1/3	HTTP OK: HTTP/1.1 200 OK - 11192 bytes in 0.000 second response time
	Memory Usage	OK	24-06-2022 18:56:17	27d 2h 30m 25s	1/3	OK - 16.8% (166588 kB) used.
	PING	OK	24-06-2022 18:56:51	27d 2h 30m 4s	1/3	PING OK - Packet loss = 0%, RTA = 0.05 ms
	Root Partition	OK	24-06-2022 18:52:25	27d 2h 29m 5s	1/3	DISK OK - free space: /var/tmp 5103 MB (64.92% inode=89%):
	SSH	OK	24-06-2022 18:52:51	27d 2h 28m 41s	1/3	SSH OK - OpenSSH_8.2p1 Ubuntu-4ubuntu0.4 (protocol 2.0)
	Total Processes	OK	24-06-2022 18:53:19	27d 2h 28m 7s	1/3	PROCS OK: 126 processes
Nagios Server 1	Current Load	OK	24-06-2022 18:53:43	49d 13h 31m 2s	1/3	OK - load average per CPU: 0.00, 0.01, 0.04
	Current Users	OK	24-06-2022 18:53:51	49d 13h 31m 2s	1/3	USERS OK - 0 users currently logged in
	HTTP	OK	24-06-2022 18:55:05	49d 13h 34m 47s	1/3	HTTP OK: HTTP/1.1 200 OK - 11192 bytes in 0.000 second response time
	Memory Usage	OK	24-06-2022 18:55:25	48d 10h 38m 36s	1/3	OK - 26.7% (531772 kB) used.
	MySQL - No. of aborted connections because of client died per sec.	OK	24-06-2022 18:55:51	49d 1h 26m 16s	1/3	OK - 0.00 aborted connections/sec
	MySQL - No. of currently running threads	OK	24-06-2022 18:56:43	49d 1h 27m 25s	1/3	OK - 2 running threads
	MySQL - Query Cache Hitrate	OK	24-06-2022 18:51:50	13d 1h 44m 46s	1/3	OK - query cache hitrate 0.00% (because it's turned off)
	MySQL - Slow Queries	OK	24-06-2022 18:52:38	45d 21h 38m 14s	1/3	OK - 0 slow queries in 320 seconds (0.00/sec)
	MySQL - Thread Cache Hitrate	OK	24-06-2022 18:52:59	45d 22h 23m 55s	1/3	OK - thread cache hitrate 97.50%
	MySQL - Threads Connected	OK	24-06-2022 18:53:23	45d 21h 37m 50s	1/3	OK - 1 client connection threads
	MySQL - Uptime	OK	24-06-2022 18:55:38	0d 0h 6m 38s	1/3	OK - database is up since 16 minutes
	PING	OK	24-06-2022 18:54:01	45d 22h 12m 54s	1/3	PING OK - Packet loss = 0%, RTA = 0.04 ms
	Root Partition	OK	24-06-2022 18:55:14	49d 13h 32m 58s	1/3	DISK OK - free space: /var/tmp 3965 MB (50.45% inode=86%):
	SSH	OK	24-06-2022 18:55:33	49d 13h 32m 4s	1/3	SSH OK - OpenSSH_8.2p1 Ubuntu-4ubuntu0.4 (protocol 2.0)
	Total Processes	OK	24-06-2022 18:56:00	45d 21h 28m 17s	1/3	PROCS OK: 134 processes
Nagios Server 2	Current Load	OK	24-06-2022 18:56:12	13d 3h 18m 21s	1/4	OK - load average per CPU: 0.00, 0.01, 0.01
	Current Users	OK	24-06-2022 18:55:01	13d 3h 17m 58s	1/4	USERS OK - 0 users currently logged in
	HTTP	OK	24-06-2022 18:54:56	13d 3h 17m 45s	1/4	HTTP OK: HTTP/1.1 200 OK - 11192 bytes in 0.000 second response time
	Memory Usage	OK	24-06-2022 18:53:11	13d 3h 16m 15s	1/3	OK - 8.1% (161560 kB) used.
	PING	OK	24-06-2022 18:56:34	13d 3h 17m 37s	1/4	PING OK - Packet loss = 0%, RTA = 0.04 ms
	Root Partition	OK	24-06-2022 18:56:39	13d 3h 17m 32s	1/4	DISK OK - free space: /var/tmp 4969 MB (63.22% inode=87%):
	SSH	OK	24-06-2022 18:54:24	13d 3h 15m 58s	1/3	SSH OK - OpenSSH_8.2p1 Ubuntu-4ubuntu0.4 (protocol 2.0)
	Total Processes	OK	24-06-2022 18:55:47	13d 3h 19m 25s	1/4	PROCS OK: 124 processes
Webserver	Current Load	OK	24-06-2022 18:54:47	0d 0h 16m 36s	1/4	OK - load average: 0.00, 0.01, 0.00
	Current Users	OK	24-06-2022 18:56:08	13d 3h 18m 37s	1/4	USERS OK - 1 users currently logged in
	HTTP Site	OK	24-06-2022 18:56:51	0d 0h 0m 6s	1/4	HTTP OK: HTTP/1.1 200 OK - 11192 bytes in 0.004 second response time
	Memory Usage	OK	24-06-2022 18:52:17	13d 3h 17m 54s	1/3	OK - 21.9% (217516 kB) used.
	PING	OK	24-06-2022 18:54:19	13d 3h 17m 27s	1/4	PING OK - Packet loss = 0%, RTA = 0.05 ms
	Root Partition	OK	24-06-2022 18:55:10	13d 3h 17m 22s	1/4	DISK OK - free space: /var/tmp 5824 MB (59.16% inode=88%):
	SSH	OK	24-06-2022 18:53:39	13d 3h 16m 12s	1/3	SSH OK - OpenSSH_8.2p1 Ubuntu-4ubuntu0.3 (protocol 2.0)
	Total Processes	OK	24-06-2022 18:54:38	13d 3h 17m 17s	1/4	PROCS OK: 140 processes
Zabbix Server	Current Load	OK	24-06-2022 18:55:17	13d 3h 17m 12s	1/4	OK - load average per CPU: 0.00, 0.01, 0.01
	Current Users	OK	24-06-2022 18:55:55	13d 3h 19m 21s	1/4	USERS OK - 0 users currently logged in
	HTTP	OK	24-06-2022 18:56:25	13d 3h 18m 59s	1/4	HTTP OK: HTTP/1.1 200 OK - 11192 bytes in 0.001 second response time
	Memory Usage	OK	24-06-2022 18:56:04	13d 3h 18m 33s	1/3	OK - 34.5% (688340 kB) used.
	PING	OK	24-06-2022 18:54:42	13d 3h 18m 12s	1/4	PING OK - Packet loss = 0%, RTA = 0.04 ms
	Root Partition	OK	24-06-2022 18:54:33	13d 3h 17m 49s	1/4	DISK OK - free space: /var/tmp 4665 MB (59.34% inode=92%):
	SSH	OK	24-06-2022 18:52:47	13d 3h 16m 50s	1/3	SSH OK - OpenSSH_8.2p1 Ubuntu-4ubuntu0.4 (protocol 2.0)
	Total Processes	OK	24-06-2022 18:55:29	13d 3h 17m 7s	1/4	PROCS OK: 169 processes

Zabbix Web Interface



Global view

All dashboards / Global view

System information

Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Number of hosts (enabled/disabled)	5	5 / 0
Number of templates	296	
Number of items (enabled/disabled/not supported)	598	550 / 0 / 48
Number of triggers (enabled/disabled [problem/ok])	327	327 / 0 [2 / 325]
Number of users (online)	5	1
Required server performance, new values per second	7.98	

5 Available	0 Not available	0 Unknown	5 Total

0 Disaster	0 High	0 Average	0 Warning	2 Information	0 Not classified

Problems

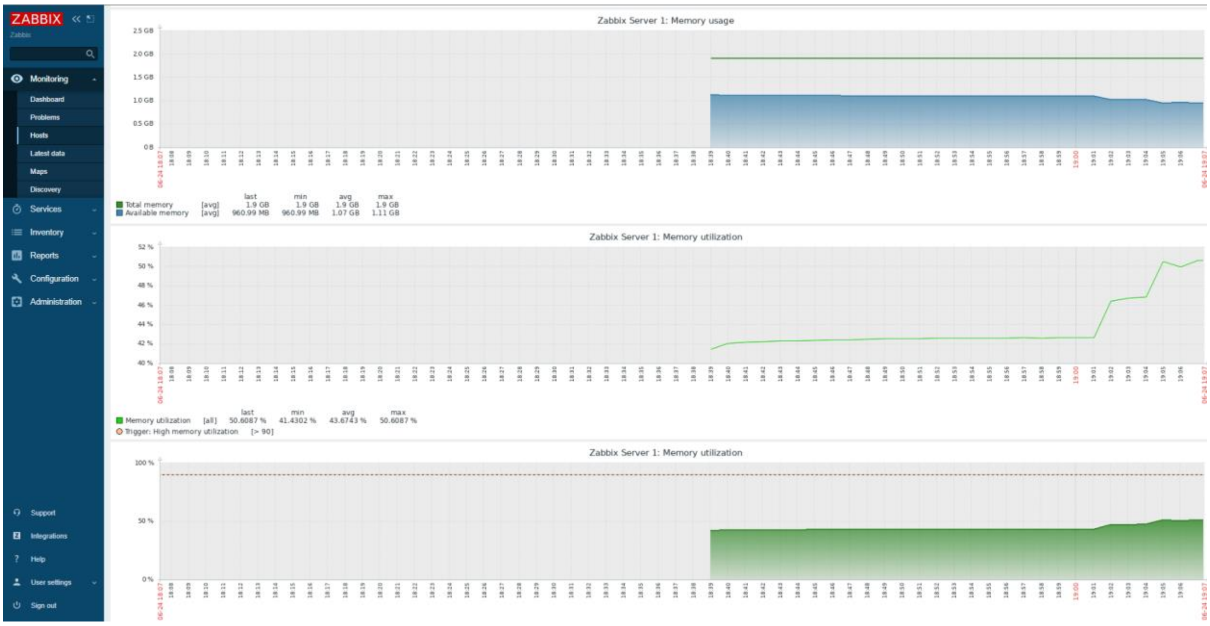
Time ▾	Info	Host	Problem • Severity	Duration	Ack	Actions	Tags
18:40:04		Nagios Server 2	Operating system description has changed	23m 5s	No		class: os component: os scope: notice •••
Today							
2022-06-11 16:33:25		Nagios Server 1	Operating system description has changed	13d 2h 29m	No		class: os component: os scope: notice •••

Favorite graphs

No graphs added.

Map

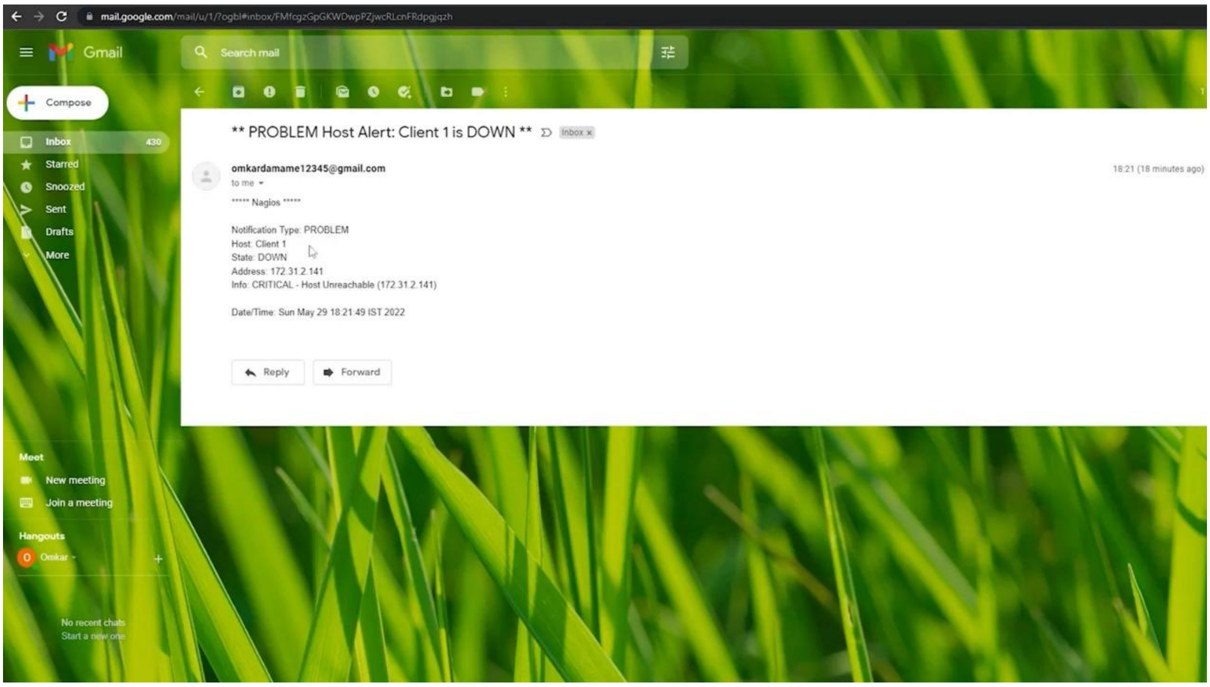
Monitoring by Graphs in Zabbix



		last	min	avg	max
🟩 Total memory	[avg]	1.9 GB	1.9 GB	1.9 GB	1.9 GB
🟦 Available memory	[avg]	960.99 MB	960.99 MB	1.07 GB	1.11 GB

		last	min	avg	max
🟩 Memory utilization	[all]	50.6087 %	41.4302 %	43.6743 %	50.6087 %
○ Trigger: High memory utilization	[> 90]				

Email notification alert by Nagios/Zabbix



** PROBLEM Host Alert: Client 1 is DOWN **

omkardamame12345@gmail.com
to me

18:21 (18 minutes ago)

***** Nagios *****

Notification Type: PROBLEM
Host: Client 1
State: DOWN
Address: 172.31.2.141
Info: CRITICAL - Host Unreachable (172.31.2.141)

Date/Time: Sun May 29 18:21:49 IST 2022

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.