Comparison of System and Network Monitoring Tools

Vidya.J[1] Prof.Shantala.C.P[2]
[1]PG Scholar [2]Professor and HOD
[1,2]Department of Computer Science and Engineering
[1,2]Channabasaveshwara Institute of Technology Gubbi, Tumkur, Karnataka, India

Abstract— DevOps is an emerging trend which mainly concentrates on communication, collaboration and integration between the developers and sys-admins. Both the team members will work together in each stage of the production lifecycle. These tools can be classified as build and test, configuration tools, monitoring tools and log monitoring tools. This paper provides introduction and comparison of some of the monitoring tools. The monitoring tools can be classified as system monitoring tools and network monitoring tools.

Key words: System Monitoring, Network Monitoring

I. INTRODUCTION

Monitoring tools are used by the developers to check for the correctness of the deployed software [1]. After collecting and analyzing the data it will be difficult to deal with performance issues, errors etc., so monitoring tools are required to deal with such issues. Some of the resources like CPU utilization, memory utilization, network bandwidth, services and so on can be monitored using these monitoring tools. System monitoring tools are responsible for collecting the data, store the collected data and based on the demand it produces graphs of the same. It is used to monitor the performance of the system. To monitor the network and its devices, the network monitoring tools are required. It will check the status of the devices and if there is any problem that has been raised then it will be notified to the administrator using these network monitoring tools, and it is also capable of scheduling the downtime. The next sections will discuss some of the system monitoring tools, network monitoring tools and its comparison.

II. SYSTEM MONITORING TOOLS

This section describes some of the system monitoring tools.

A. Graphite System Monitoring Tool

Graphite is an open source graphing tool, which can be used to track the values of the metrics that varies according to time. It also measures the performance of the computer [3]. It is responsible for collecting the metrics, storing the collected metrics and it will produce the graphs based on the demand. These are the components of Graphite [2]: Carbon is the processing backend which will receive the data. As the Graphite cannot collect the data by its own, it needs some other tools to accomplish this task. The Carbon will store the received data in the database called Whisper. The data can be viewed through the frontend Graphite webapp, which is responsible for producing the graphs depending on the user request.

B. Ganglia System Monitoring Tool

It is an open source tool for monitoring the grids and clusters [4]. All the nodes of the cluster use multicast address to share the data with other nodes, this information will be maintained by all the nodes so that in case of any crash this information is used for recovery. To represent the data XML is used, for data transport XDR is used and it uses RRD tool for storing the data. These are the components of Ganglia [5]: Ganglia monitoring daemon will be installed on all the devices that has to be monitored and it interacts with the OS of the system to collect the system metrics. Ganglia meta daemon is responsible for collecting the data from different gmond sources or from other gmetad sources and the collected data will be stored in the round-robin database. The PHP web frontend is used to represent the data.

C. Sensu Monitoring Tool

It is an open source monitoring tool which can be used in the cloud environment [6]. This is also referred to as a monitoring router that will perform checks to know whether the service/device is up/down and it produces the exit status codes to indicate the status of the device. This result is passed to the handlers to take respective actions. The following are the components of Sensu [7]: the server is responsible for initiating the checks on the clients, where these clients are installed on all the monitored nodes. Redis database is used to store the data. For exchanging the messages between the server and the client RabbitMQ is used. In order to access the data stored in the Redis database REST API is used, this API is also responsible for doing some tasks like resolving the events, it will issue check execution request and so on.

III. NETWORK MONITORING TOOLS

This section provides a brief introduction of some network monitoring tools.

A. Zabbix Network Monitoring Tool

It is an open source monitoring tool used to monitor different parameters of the network services and other devices that belong to the network. The status of the networked devices can be accessed from the remote location also and it is used by small as well as large organizations [8]. These are the components of Zabbix [9]: Agents are installed on all the nodes to be monitored and it reports the data to the server, where this server is a central repository that collects the data from all of its agents. The collected data is stored in the database. The server as well as the web interface will be located on the same machine. Zabbix proxy is optional; it can be used to distribute the load so using proxy is advantageous to deal with heavy load.

B. Nagios Network Monitoring Tool

It is an open source monitoring tool used for monitoring, scheduling and alerting [10]. It is responsible for notifying the administrator if any problem is identified. This tool will make use of plugins to accomplish any task. These plugins can also be executed on the remote machines using some of the agents like NRPE. It is capable of scheduling downtime. It will check the services, process or devices and returns the

exit status code. These are the components of Nagios [11]: Plugins (it will add a new functionality to the existing software) are the main components of Nagios. These are used to check the services and returns result to the server which takes the corresponding actions, results will be displayed on GUI. The GUI will display alerts, graphs, configuration and so on.

C. Cacti Network Monitoring Tool

It is an open source monitoring tool which is capable of collecting the data, storing the collected data and it displays the graphs of the same [12]. It is designed as a frontend application. This performance monitoring tools uses Linux/Apache/MySQL/PHP (LAMP) stack and RRD tool. It will poll the routers or switches in order to measure the network traffic. By polling the nodes it will collect the data or it uses SNMP to collect the data from the remote hosts. The collected data will be stored in the MySQL and RRD tool [13]. This data can be visualized through any browser; the RRD tool will provide a built-in function for producing the graphs. It provides templates which saves time.

IV. Comparison of Monitoring Tools

The Table 1 shows the comparison of the three system monitoring tools Graphite, Ganglia and Sensu. Among the three system monitoring tools Graphite is considered as a best tool if it is combined with collectD/statsD (it can collect the data) [15, 16], because of the following reasons: It will represent the data in a better way compared to other tools, the main advantage of using Graphite is its specialized database called Whisper-which is similar to the RRD tool but possess some extra functionalities like it allows the user to insert multiple data points at a time i.e., in a single write operation; it will help to improve the performance of the system even under the heavy load [2]. Always real-time graphs will be produced, i.e., if the data is not present in the disk then the frontend will take the data directly from the backend in order to produce graphs [2, 14]. It supports data retention also where the retention limit is specified by the user.

Features/Tools	Graphite	Ganglia	Sensu
Open source	Yes	Yes	Yes
Written in	Carbon and Whisper-Python; Graphite webapp-JavaScript	Gmond and gmetad-C and Python; Frontend-PHP and JavaScript	Ruby; Redis-C
Platform	UNIX/Linux	Cross platform	Ubuntu, CentOS
Cloud environment	Yes	Yes	Yes
Data retention	Retention limit is specified by the user	1-3 years	Retention limit is specified by the user
Data collection	Whisper database	RRD tool	Redis database
Data visualization	Yes	Yes	Yes
Distributed monitoring system	No	Yes	No
Information sharing	No	Yes	No
Exit status codes	No	No	Yes
Notification/Alerting	No	No	Yes
Checks	No	No	Yes
Handlers	No	No	Yes
Scalable	Yes	Yes	Yes
Secure	Yes	Yes	Yes
Integration	Yes	Yes	Yes

Table 1: Comparison of System Monitoring Tools.

Features/Tools	Zabbix	Nagios	Cacti
Open source	Yes	Yes	Yes
Written in	Server components, agents, proxy-C; Frontend-PHP and Java	C language; Plugins can be written in any languages	PHP, RRDtool-C
Platform	Cross platform	Cross platform	Cross platform
Data collection	MySQL/Oracle/SQLite	Flat files/SQL database	Flat files/MySQL/RRD
Notification/Alerting	Yes	Yes	No
Plugin based architecture	Yes	Yes	Yes
Polling	Yes	No	Yes
Trapping	Yes	No	No
Log monitoring	Yes	Yes	Yes
Logical grouping	Yes	Yes	Yes
Distributed monitoring system	Yes	Yes	Yes
Proactive monitoring	Yes	Yes	Yes
Redundant monitoring	Yes	Yes	Yes
Automatic network	Yes	Yes	Yes

discovery			
Agentless monitoring	Yes	Yes	Yes
Adaptive monitoring	No	Yes	No
Access control	Yes	No	Yes
Availability monitoring	Yes	Yes	No
Auto padding	No	No	Yes
Third party tools	Yes	Yes	No
Templates	Yes	Yes	Yes
Topology	No	Yes	No
Scheduling downtime	No	Yes	Yes
Scalable	Yes	Yes	Yes
Integration	Yes	Yes	Yes

Table 2. Comparison of Network Monitoring Tools.

The Table 2 shows the comparison of three network monitoring tools Zabbix, Nagios and Cacti. Among the three network monitoring tools Zabbix can be considered as the best tool. Its reporting is very flexible. Some of the features of Nagios (Availability monitoring) and Cacti (graphing) is combined in the Zabbix [17]. It uses MySQL structure which is easy to understand, though it consumes much space for storing the historical logs it can be fine-tuned [18]. The agents will be installed on all the nodes, which support resource monitoring features. The graphs produced by this tool are very flexible and extremely good. Use of templates will save lot of time [19]. These are some of the features supported by Zabbix [18]: auto registration, aggregate graphs, polling, trapping and so on.

V. CONCLUSION

These monitoring tools are used by the developers to know whether the deployed software is working properly, as expected or not. According to the current survey, Graphite and Zabbix are considered as the best tools among the above discussed system and network monitoring tools. These system monitoring tools are capable of collecting the data, storing the collected data and produces graphs of the collected data. These network monitoring tools are capable of notifying the administrator in case if any problem arises.

REFERENCES

[1] http://blogs.vmware.com/cloudops/tag/devops
[2] Chris Davis, Graphite Documentation Release 0.10.0, October 18, 2014,
[3] http://graphite.readthedocs.org/en/latest/release.html
[4] http://en.wikipedia.org/wiki/Graphite_%28software%29
[5] https://en.wikipedia.org/wiki/Ganglia_(software) [5] www.ganglia.sourceforge.net
[6] https://sensuapp.org
[7] https://sensuapp.org/docs/latest/overview
[8] https://en.wikipedia.org/wiki/Zabbix
[9] https://www.zabbix.com/documentation/2.0/manual/introduction/about
[10] https://en.wikipedia.org/wiki/Nagios
[11] Afeez Yusuff, "Network Monitoring: Using Nagios as an example tool", May 2012.
[12] https://en.wikipedia.org/wiki/Cacti_(software)
[13] www.cacti.net/downloads/docs/html/operating_principles.html
[14] www.aosabook.org/en/graphite.html
[15] http://graphite.wikidot.com/whisper
[16] http://stackoverflow.com/questions/11179622/performance-monitoring-ganglia-vs-graphite
[17] https://workaround.org/article/tired-of-nagios-and-cacti-try-zabbix
[18] http://www.quora.com/Which-is-the-best-network-monitoring-tool-Nagios-or-Zabbix
[19] http://kkovacs.eu/zabbix-vs-nagios

