

# Developing Integrated Smartphones Notification of Server Resource Monitoring System Using Zabbix, Webhook, and Telegram

Sulasno<sup>1</sup>, Rakhmat Saleh<sup>2</sup>, Intan Savitri<sup>3</sup>

<sup>1,2,3</sup>Center for Empowerment of Informatics and Nuclear Strategic Area, National Nuclear Energy Agency, Indonesia

<sup>1</sup>sulasno@batan.go.id, <sup>2</sup>rakhmats@batan.go.id, <sup>3</sup>intan\_savitri@batan.go.id

**Abstract** - Server monitoring activities are important to ensure the availability of resources used for the execution of its internal processes. Generally, Administrators monitor the use of server resources through a computer using a browser. The problem with monitoring through a computer is less flexible and slow for checking notifications about the state of problem server resources in real-time anytime and anywhere using smartphones. This study aims to develop notification integration on smartphones from a server resource monitoring system using Zabbix Webhook and Telegram. The research method used is the Cisco PPIDIO Lifecycle Approach to Network Design and Implementation. As a case study, it is located at the Center for Empowerment of Informatics and Nuclear Strategic Area, the National Nuclear Energy Agency. The results of the study are notifications on smartphones via the Telegram bot account indicating a problem with resource usage. The results of this study have been used by server Administrators. The server Administrators have provided information about the benefits of this system, for flexibility, speed of getting information in resource usage, identifying, and resolving problems before there are complaints so that the quality of service is getting better.

**Keywords:** smartphones notification, server monitoring, Zabbix Webhook, Telegram

## I. INTRODUCTION

The dependence on using information technology (IT) to support an organization's business processes demands the high availability of IT services. The dependence on IT, using through the various information system applications running on various servers. For example, the Center for Empowerment of Informatics and Nuclear Strategic Area, the National Nuclear Energy Agency, currently manages 58 servers. These servers are used to run information systems, both information systems which are internal applications and public service applications for the wider community. Various applications running on servers will store data at any time with high data growth. Besides that, users from

internal and from the public in accessing public services need excellent service. Therefore, we need servers that perform reliably as a place for application execution, so that they can work effectively and efficiently, so it is necessary to monitor the availability of resources in real-time at any time.

The Center for Empowerment of Informatics and Nuclear Strategic Area already has a server resource monitoring system developed using the Zabbix 4.0 software. In this system, the Administrator monitors server resources through a work computer. From the existing system, there is still the problem of not being flexible and fast to receive notification of resource conditions, because to check notifications, Administrators cannot use smartphones that can be accessed anywhere and anytime and almost every Administrator has them. In addition, the system used now uses Zabbix 4.0 with support for a personal home page programming language (PHP) 5.4.0, where the Zabbix release software does not yet support webhook for sending notifications based on smartphones, especially using the Telegram bot. In addition to, above problems, the software used for operations such as operating systems and software for application development uses open source and free based software. Such an operational environment provides many advantages, because in addition to budget savings, increasing the creativity of personnel, the existing budget can also be used to increase human resource capabilities or expenditures for much-needed IT equipment. Therefore, the appropriate solution to the problem raised is to optimize the system used through the development of integrated notifications to inform that critical state of resources using flexible smartphones and a popular chat application, namely Telegram, so that it is easily accessible anytime and anywhere. Furthermore, the old system needs to be upgraded to a system that supports webhooks for sending notifications on smartphones via the Telegram bot service. Here we explain a little about

many terms we use, namely Telegram, Webhook, System Resources, System Monitoring, and Zabbix.

Telegram was initialized in August 2013 by Russian-born businessman Pavel Durov. Telegram is one of the popular messaging services which is based on an open-source platform and is a free service without payment. The Telegram application can send various kinds of information such as images, video, audio, and location to users. According to the Statista survey, this number of telegram users is increasing from year to year, and the number of monthly active users in March 2019 is around 200 million [1]. This information illustrates that Telegram can be argued as one of the most popular messaging services in the world. Telegram is equipped with an Application Programming Interface (API) that allows users to create chatbots [2]. Chatbots are accounts without a phone number that can handle messages automatically via a set of commands. While Webhook is an API concept that is currently popularly used. One method for creating a Telegram bot through software is a webhook. This method is used to determine the uniform resource locator (URL) and receive incoming updates. Whenever there is an update for a Telegram bot it will send a POST HTTPS request to a specified URL. Zabbix version 5.0 includes properties to support a webhook. Currently, most of the popular applications such as WhatsApp, Facebook Messenger, WeChat, Line, Telegram, provide chatbots that are served based on the webhook concept. It is a transmission method to redirect activity to a registered call-back URL when an event occurs between client and server [3]. Zabbix webhook integration uses special JavaScript code to make HTTPS calls. This allows for integration with external systems such as a Telegram bot.

Computer system resources are physical or virtual components of the limited availability of devices in a computer system. Devices connected to the computer system are a resource. Every internal component is a resource. Virtual system resources include files, network connections, memory area, central processing unit (CPU) time, hard disk space, and other internal supporting devices [4]. Availability of server resources is needed to maintain server performance to remain reliable [5-6]. A monitoring system can be defined as an element that is implemented for targeting network devices or servers. This activity is carried out periodically to check about availability and performance status of each node and link. In case there is a problem, or some elements are not available, the monitoring system will automatically notify the responsible Administrator. In some cases, it is possible to actively manage the network or server using a monitoring system. We can also define effective and

efficient ways that will be used to detect existing cases or problems [7-8].

The last terms is Zabbix. Zabbix is one of the software that is widely used for the implementation of a server resource monitoring system or network devices [9]. Of course, many criteria must be met in choosing monitoring software, but the advantages of this software are open-source and free, graphical interface, and use structured query language (SQL) database as data manager [10-11]. Besides that, Zabbix is a multiplatform-based software used to monitor the availability of server resources and network equipment infrastructure with a distributed monitoring system. Zabbix can generate graphical-based statistical reports, screen monitoring, and notifications via email or Telegram if a device has a problem. Zabbix must be installed and configured. In this study, there are two main tools for the Zabbix architecture used, namely the Zabbix Server and the Zabbix Agent. Zabbix Server is the main process of the Zabbix software installed on the main server as a monitor. The Zabbix Agent is a UNIX daemon that runs on the host (server) that is being monitored. This agent provides information on the state of use monitored host resources and sends this information to Zabbix Server. Protocols for communicating in the Zabbix system are very simple, including transmission control protocol/internet protocol (TCP/IP) or simple network management protocol (SNMP) which are generally used on network devices and servers. Another advantage of Zabbix is that Administrators can monitor resources through a graphical dashboard based on the graphics and is easy to configure [12].

Several studies related to this paper have been carried out. As in [13] notification research has been conducted between Zabbix and Telegram. But in this research, Zabbix is used for monitoring active network devices such as Mikrotik which uses the SNMP protocol to connect network devices with the Zabbix Server. Furthermore, in [1] a chatbot was developed as an addition to notification via Telegram and providing information through notification of disaster events in Japan to the public. The research is not related to monitoring IT resources. Research [2] is carried out by building a remote control device with Esp 8266 to access local device control through the help of an artificial intelligence Chatbot using Telegram messenger. The system controls electronic devices in the workspace via the internet, such as lights, fans, drawer locks, power outlets, and temperature checks. This research is not related to monitoring server or network resources. In the

research activity [3] the messenger of anti-virus system was integrated that utilizes chatbot services to predictively detect malware attacks targeting messengers. This system integrates various chatbot messengers into one form of adapter so that it can support all messengers and contains a multi-antivirus scanner tasked with detecting malware, so users can check the detection results in real-time. This activity is not related to monitoring the use of a server or network resources in real-time. Furthermore, in research [14], SQL injection logs captured when hackers try to access the webserver are sent directly to the Telegram account and then forwarded to the Administrator's smartphone. Sending logs to the telegram bot is done through a script. This research is not related to monitoring the use of a server or network resources because what is being monitored is related to attacks on these servers from the network. Finally, in research [15], a monitoring system was developed for server resources using Zabbix, but in this activity, it has not been integrated with Telegram in receiving notifications about the state of monitored server resources.

This study aims to develop notification integration on smartphones through a server resource monitoring system currently in use. The system is designed, installed, and configured as well as the implementation of the JavaScript program to be able to send critical state notifications from the results of real-time monitoring of server resource usage. When there are constraints on server resource usage then the monitoring system will notify a Telegram bot group Administrator account. After the monitoring system is integrated with smartphone notifications, the system must still display a web-based monitoring dashboard that is accessed via the intranet. Monitored server resources include hard disk quotas, CPU utilities, memory usage, and the amount of data packet traffic for ethernet servers and others. Monitored servers must have Zabbix Agent installed on them.

## II. METHOD

This study uses the PPDIIO Lifecycle Approach to Network Design and Implementation model [16]. This model was developed by Cisco and has been widely used in the development and management of computer networks [17-18]. This method has six stages which be described in Fig. 1.

### A. Prepare

The preparation stage describes the need to develop a computer network system that is used. Next, the technology is needed for system development, including

hardware or software requirements. The preparation stage in this research contains the need for developing integrated notification on smartphones from the server resource monitoring system that is being used. This activity includes collecting information about notifications based on smartphones from the server or network monitoring system using messaging media such as Telegram, WhatsApp, and others.

### B. Plan

The planning stage contains an identification of network requirements, other facilities used, and user requirements based on the objectives of the system to be developed. Besides that, identification of the characteristics of a network system is carried out to assess the system. In this study, the identification of the system that is running, the requirements for some components of the system to be developed is carried out. Next, identify the technological devices that are planned to be used for the development of integrated smartphone notifications.

### C. Design

System design is developed by technical requirements of the previous system conditions. The network design specification is a comprehensive design that meets technical requirements. The design results include a network diagram and a list of equipment used. In this study, a system architecture design was developed based on the system being used. The architectural design also shows communication between the devices used.

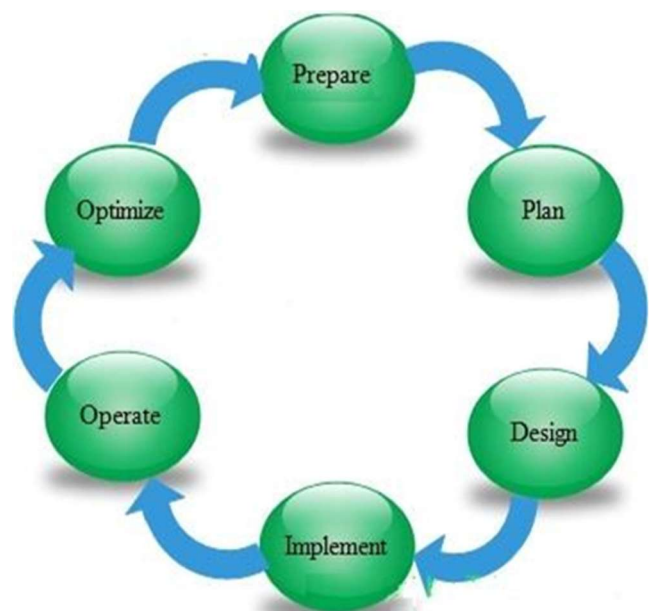


Fig. 1 PPDIIO development model

*D. Implement*

At this stage, the equipment as part of the developed system components is installed and configured. Installation and configuration are carried out by the architectural design that was made in the previous stage. The implementation in this study starts from the activity of upgrading the Zabbix Server software and the PHP programming language to webhook configuration and JavaScript implementation. The webhook configuration and JavaScript implementation are carried out on the Zabbix Server software.

*E. Operate*

The operational phase is the daily operational activities of the system being developed. Operational activities include management activities, monitoring of developed system components, managing performance, conducting and correcting errors that occur. In this study, operational monitoring system activities were carried out either through a computer browser for web monitoring dashboards or through checking notifications on smartphones. System operation is carried out to prove that the installation and configuration that have been done are correct. If there is a mismatch, a correction is made of the configuration errors that have occurred.

*F. Optimize*

The optimization phase involves the awareness of the management/system Administrator to identify and fix problems before they disrupt the operational system. The optimization phase allows for system modifications if too many problems arise. The requirements for the modified system design direct the development of the system to the start of the cycle from the existing phase at PPDIIO.

(HCI) system with hardware and software specifications on the virtual server is as presented in Table I. The server monitors the use of server resources totaling 37 servers. Before this system was created, the Administrator has been monitoring through a browser using the local internet protocol (IP) from the server. Furthermore, so that the server can provide notifications to smartphones using the Telegram bot messaging application, it is necessary to upgrade the Zabbix software and PHP server to the latest released.

The next activity is to design the system architecture being developed. The architectural design is shown in Fig. 2 [3], [12]. In general, the architecture consists of four parts, namely Zabbix Server (ZS), Zabbix Agent (ZA), Administrator Smartphone (AS), and Telegram Server (TS). ZS is the main server for monitoring resource usage. ZS is the core tool of the resource usage monitoring system. ZS monitors all servers that are monitored based on reports from ZA. Furthermore, ZA is a server device that is monitored for the state of its resources. To monitor ZA the Zabbix-Agent software must be installed and configured on it. Furthermore, AS is the smartphone device that is used by the Server Administrator to receive notifications of resource usage state if there is a problem. Administrators can also configure trigger and monitoring response times via ZS to be sent to ZA as well as critical state notifications sent to the AS. If there is a critical condition according to the configuration, ZS will send a notification to the bot's Telegram account which is sent to the AS device via TS. TS is a Telegram server service that serves messaging applications and APIs. Administrators via smartphones can communicate to TS via standard Telegram commands for creating bots and group chats.

III. RESULTS AND DISCUSSION

*A. Developing System*

Integration of notifications on smartphones from a running server resource monitoring system using the Zabbix webhook and Telegram bot will be explained in this section. The first activity starts from collecting data on the server resource monitoring system used. The server resource monitoring system that runs using a virtual server on the hyper-converged infrastructure

TABLE I  
SPECIFICATION OF SERVER MONITORING

No	Name of Devices	Amount
1	Processor	4 Core
2	Memory	16 Gb
3	Hard Disk	400 GB
5	Operating System	Centos 7.7
6	Monitoring Software	Zabbix 4.0
7	Programming Language	PHP 5.4.0

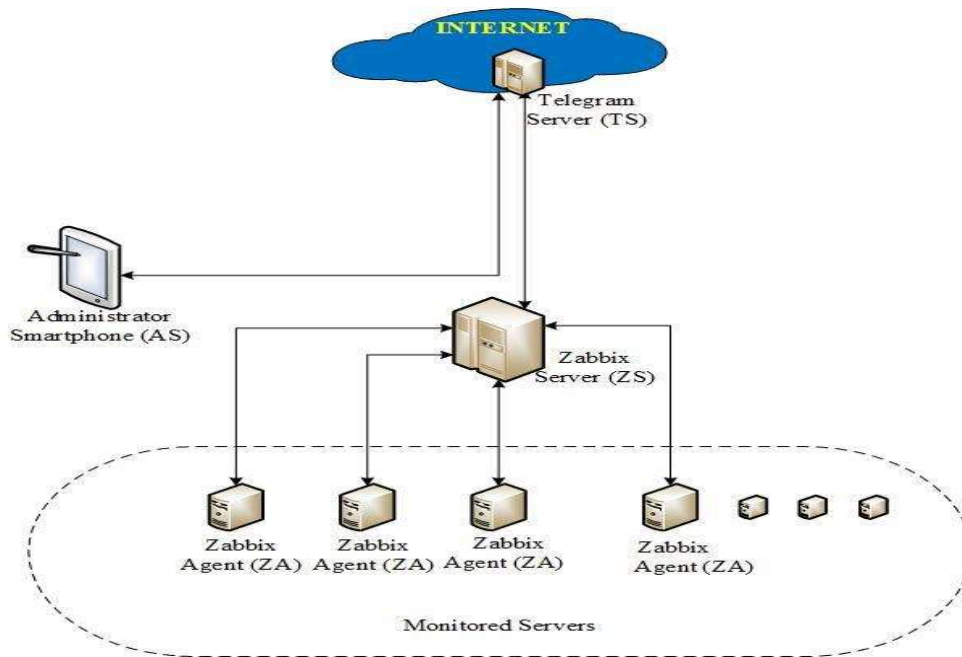


Fig. 2 Architectural design

The next activity is system upgrade. The system upgrade activity was carried out by updating the ZS software and PHP programming language to be released the latest on ZS. This is done because the ZS software that is currently used on a running system does not yet support the use of webhook. Furthermore, the ZS software to be executed requires the latest released PHP programming language. On this occasion the ZS software was upgraded to released 5.0 and PHP was upgraded to release 7.2. Upgrading is done with upgrade/installation commands for both ZS and PHP software. Meanwhile, the server operating system used remains unchanged, namely Linux Centos Release 7.7.

Then the activity of creating a Telegram bot account was carried out. There are two kinds of making a Telegram bot account, namely an account for the Administrator bot ZBXsermonbot which is edited and converted to ZBX-B4T4N, and an account for the Administrator group, namely ZBX-BAT4N-GROUP. The ZBX-B4T4N account will later become a member of the ZBX-BAT4N-GROUP account. The activity of creating a Telegram bot account ends with the identification of the Telegram token for ZBXsermonbot and the Telegram group ID which will later be used in

the Zabbix webhook configuration. Account creation is done via AS devices and web-based Telegram.

The next job is to configure the Zabbix webhook on the media type in ZS and implement the JavaScript program. The media type display for the webhook configuration on ZS is presented in Fig. 3. From this figure, it can be argued that the important parameters that must be filled in according to the data generated on the Telegram bot are Message, ParseMode, Subject, TelegramToken, and To. Message value filled with the desired message will appear at the end of the notification on the Telegram bot chat. ParseMode is filled with Markdown, Subject is filled with {ALERT.SUBJECT}. Then TelegramToken is filled with tokens that have been identified from the Telegram account ZBXsermonbot and to is filled with the Telegram group ID for ZBX-BAT4N-GROUP. In addition to the configuration of the media type, the configuration is required for the user media section and triggers actions for defining the user and monitoring the state of the resource so that notification is required. Due to security reasons, Token ID and Group ID are not shared with the public and are displayed in this paper.

The screenshot shows the Zabbix configuration interface for a Telegram Webhook. The form is organized into sections: 'Name' (Telegram Webhook), 'Type' (Webhook), 'Parameters' (a table with columns Name, Value, and Action), 'Script' (try {...}), and 'Timeout' (10s). The Parameters table lists: Message (Alert message from SERMON), ParseMode (Markdown), Subject ({ALERT.SUBJECT}), telegramToken ({TELEGRAM.TOKEN}), and To ({ALERT.TO}). Each parameter has a 'Remove' button next to it. There is also an 'Add' button at the bottom of the Parameters table.

Fig. 3 Zabbix configuration

The next activity is about the implementation of the JavaScript program as shown in Fig. 4. The implementation of the JavaScript program is carried out because the new system is an upgrade from the old system that is being used. As shown in Fig. 4, the chatid variable is filled with the telegram ID of the ZBX-BAT4N-GROUP Administrator group. Then the botid variable is filled with the Telegram token from the ZBXsermonbot account.

#### B. Server Monitoring and Smartphone Notification

At the operational stage, the server Administrator monitors the use of server resources through a browser on a desktop computer and through receiving smartphone notifications on the Telegram bot account for the Administrators group. The operational activities

of the system being developed are carried out to show that the system is working by the objectives of the desired system development activities. Besides that, the system being developed is also an upgrade and integration of the ongoing system, therefore the Administrator needs to monitor resource use through the ZS dashboard [19]. The ZS dashboard is accessed through a browser with an address using a local intranet. The resources monitored include hard disk usage, CPU utility, memory usage, and the size of data packets passing through the ZA ethernet server or other resources [20-21]. Fig. 5 shows some of the results of monitoring the use of hard disks, memory, CPU utilities, and data traffic passing through the ethernet server. The monitoring results in Fig. 5 are taken from resource usage on the DIGILIB-V server and the SITP-V server.

```

try {
  var params = JSON.parse(value);

  var chatid = "Telegram group ID";
  var botid = "Telegram token";

  var req = new CurlHttpRequest();
  req.AddHeader('Content-Type: application/x-www-form-urlencoded');
  var URL = encodeURI("https://api.telegram.org/bot" + botid +
"/sendMessage?chat_id=" + chatid + "&text=" +
params.Subject + "\n" + params.Message)
  Zabbix.Log(1, 'Telegram Webhook Url : ' + URL);
  var result = req.Get(URL, {});
  Zabbix.Log(1, 'Telegram Webhook Result : ' + result);
  return true
} catch (error) {
  Zabbix.Log(1, "Telegram Webhook Failed : " + error);
  throw error
}

```

Fig. 4 JavaScript program



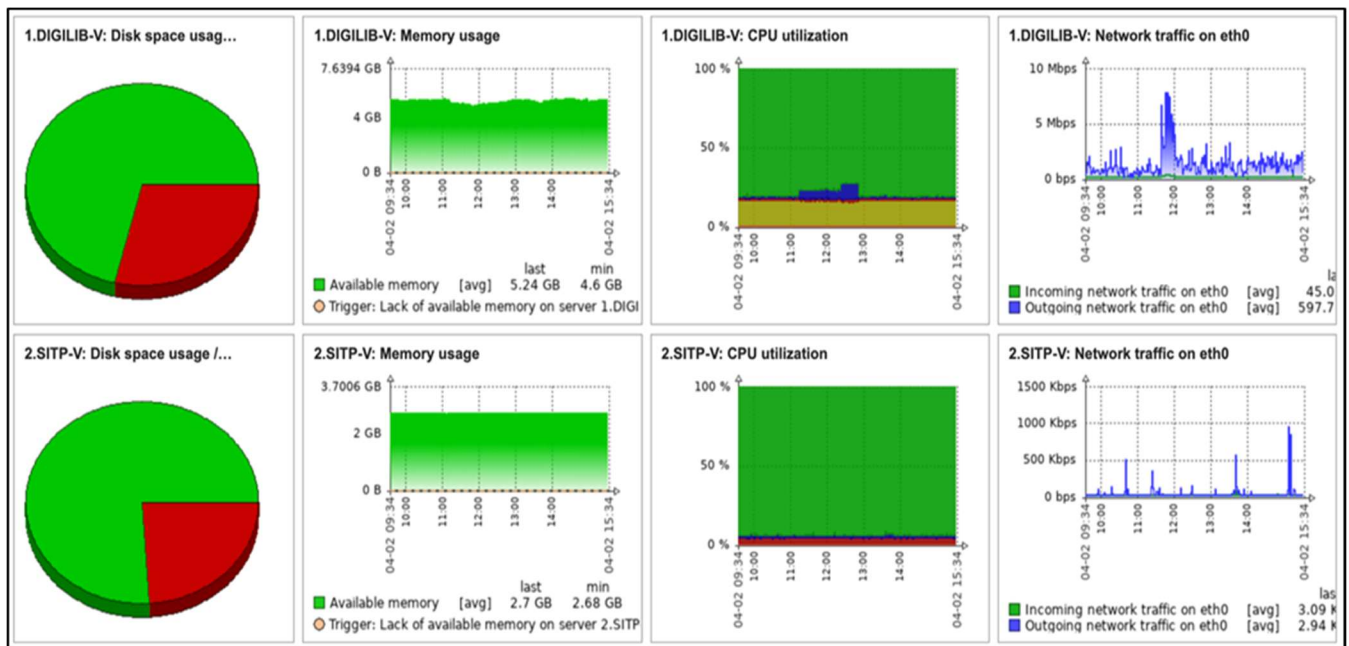


Fig. 5 Monitoring dashboard display

Next in Fig. 6, the results of the notification trial on the ZS dashboard are presented if there is a ZA server that has a problem. As a trial, a shutdown of one of the ZAs was carried out [17]. The ZA used for testing is the DIGILIB-V server by shutting down/stopping the Zabbix Agent process so that communication between ZS and DIGILIB-V cannot be implemented. After the Zabbix Agent shutdown process on the server and when ZS monitors the DIGILIB-V server, it does not find the server, then ZS displays a notification message on the dashboard as shown in Fig. 6. The experiment was carried out on April 4, 2021, at around 15:11:14. From the information displayed in Fig. 6, it can be conveyed that ZS notified of a problem on the DIGILIB-V server because the server could not be reached or communication for monitoring was carried out.

Furthermore, in Fig. 7 a notification is given from TS via the ZBX-BAT4N-GROUP Telegram group account that reaches the AS device. The contents of the notification information that there is a problem on DIGILIB-V server because it cannot be reached or recognized from ZS so that the server cannot be monitored. In general, if an actual condition occurs based on notification from TS that reaches the AS device, the server administrator can be check in the ZA device both remotely and physically to carry out repairs of existing problems. The server Administrator checks remotely first. If it cannot be done remotely, then continue to check physically by coming directly to the location of the server.

Problem hosts						
Time ▼	Info	Host	Problem • Severity		Duration	Ack Actions
15:11:14		1.DIGILIB-V	Zabbix agent on 1.DIGILIB-V is unreachable for 5 minutes		39s	No 4 →
Today						

Fig. 6 DIGILIB-V server problem message on ZS

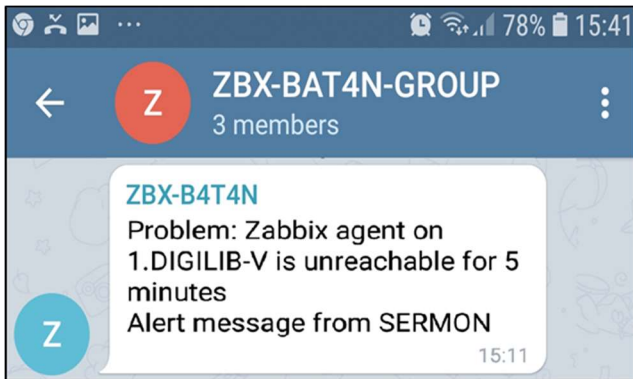


Fig. 7 DIGILIB-V server problem notification on AS

In Fig. 8, the results of the notification trial on the ZS dashboard are presented if there is a ZA server that has a problem then it is corrected. As a trial, the DIGILIB-V server repair was carried out by starting the Zabbix Agent process so that communication between ZS and ZA could be carried out. After starting the Zabbix Agent process at ZA, a notification message is displayed on the ZS dashboard as shown in Fig. 8. This experiment was carried out on April 4, 2021, at around 19:51:17. From this information displayed in Fig. 8, it can be conveyed that ZS notifies that there will be an improvement on the DIGILIB-V server because a server can be reached, or communication is made for monitoring.

Furthermore, in Fig. 9, a notification from TS is delivered via the ZBX-BAT4N-GROUP Telegram group account that reaches the AS device. The contents of the notification informing that previously problematic DIGILIB-V have been updated so this server can be

monitored by ZS. The incident reached AS equipment at 19:51.

From ZS and AS operational data shown in Fig. 5 to Fig. 9, which represents information of the operational results of ZS and AS, it can be argued that integration of notifications on smartphones from a server resource monitoring system can run well, following through development of the system have been done. The monitoring results that contain real-time server resource usage information are also displayed on a web-based dashboard that can be accessed online via a work computer using a browser application via the intranet. Furthermore, if there is a problem with the use of server resources, ZS will send a notification to TS. Next, TS will continue about notification to AS device. The monitoring system runs well if the information displayed is following through results of desired system development for monitoring resources of the servers used [20], [22]. During the operational stage, if ZS is experiencing a lack of resources such as hard disk, memory, CPU, and others, optimization can be easily carried out through additional resources because ZS uses a virtual server on the HCI system. Furthermore, the results of the integration of smartphone notifications from the server resource monitoring system developed have been operationalized to provide easy real-time monitoring of server resource usage. In addition, the Administrator adds information that with the help of this system it can work more efficiently, identify, and resolve problems before there are complaints so that the quality of service is increasing.

Time	Recovery time	Status	Info	Host	Problem • Severity	Duration	Ack	Actions
2021-04-03 15:11:14	2021-04-03 19:51:17	RESOLVED		1.DIGILIB-V	Zabbix agent on 1.DIGILIB-V is unreachable for 5 minutes	4h 40m 3s	No	17

Fig. 8 DIGILIB-V server repair message on ZS

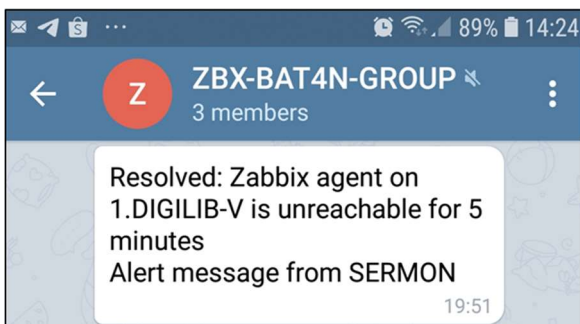


Fig. 9 DIGILIB-V server repair notification on AS

#### IV. CONCLUSION

The development activities for the integration of notifications on smartphones have been successfully carried out from the server resource monitoring system that is being used using the Zabbix webhook and Telegram. The results obtained are notifications on the smartphones Administrator server if there are problems with server resource usage. Besides, the results of monitoring the use of server resources are also displayed on a dashboard in real-time and can be accessed via the



intranet by the Administrator as proof that the old system that has been upgraded can function properly. The information displayed through notifications on smartphones and web-based dashboards make it easy for server administrators to evaluate resource usage and design the required server capacity development. The activities that have been carried out are only developing notification integration on smartphones from the monitoring system of the resources that are being used. Further research is needed to evaluate server security and smartphone device security as notification recipients using APIs.

#### REFERENCES

- [1] S. E. Ahmady and O. Uchida, "Telegram-based chatbot application for foreign people in Japan to share disaster-related information in real-time," *2020 5th Int. Conf. Comput. Commun. Syst. ICCCS 2020*, pp. 177–181, 2020.
- [2] M. Muslih, Somantri, D. Supardi, E. Multipli, Y. M. Nyaman, A. Rismawan, Gunawansyah, "Developing smart workspace based iot with artificial intelligence using telegram chatbot," *Proc. - 2018 4th Int. Conf. Comput. Eng. Des. ICCED 2018*, pp. 230–234, 2019.
- [3] S. Lee, J. Lee, W. Lee, S. Lee, S. Kim, and E. T. Kim, "Design of Integrated Messenger Anti-virus System using Chatbot Service," *Int. Conf. ICT Converg.*, vol. 2020-October, pp. 1613–1615, 2020.
- [4] K. B. Sandeep, M. Raghavendra, and K. J. Shetty, "Performance analyzer: An approach for performance analysis of enterprise servers," *2016 IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol. RTEICT 2016 - Proc.*, pp. 480–484, 2017.
- [5] P. Garraghan, P. Townend, and J. Xu, "An analysis of the server characteristics and resource utilization in Google cloud," *Proc. IEEE Int. Conf. Cloud Eng. IC2E 2013*, pp. 124–131, 2013.
- [6] A. Datt, A. Goel, and S. C. Gupta, "Analysis of Infrastructure Monitoring Requirements for OpenStack Nova," *Procedia Comput. Sci.*, vol. 54, pp. 127–136, 2015.
- [7] A. Iqbal, C. Pattinson, and A. L. Kor, "Performance monitoring of Virtual Machines (VMs) of type i and II hypervisors with SNMPv3," *2015 World Congr. Sustain. Technol. WCST 2015*, pp. 98–99, 2016.
- [8] Z. Jiang, "A Linux Server Operating System's Performance Comparison Using Lmbench," *2016 Int. Conf. Netw. Inf. Syst. Comput.*, pp. 160–164, 2017.
- [9] L. C. Mota, E. D. Moreno, A. L. Ribeiro, and R. J. P. B. Salgueiro, "A comparative analysis of network management protocols in IoT applications," *J. Comput. Sci.*, vol. 14, no. 9, pp. 1238–1246, 2018.
- [10] O. Marik and S. Zitta, "Comparative analysis of monitoring system for data networks," *Int. Conf. Multimed. Comput. Syst. -Proceedings*, pp. 563–568, 2014.
- [11] L. Macura and M. Voznak, "Multi-Criteria Analysis and Prediction of Network Incidents Using Monitoring System," *J. Adv. Eng. Comput.*, vol. 1, no. 1, pp. 29, 2017.
- [12] C. M. Petruți, B. A. Puiu, I. A. Ivanciu, and V. Dobrota, "Automatic Management Solution in Cloud Using NtopNG and Zabbix," *Proc. - 17th RoEduNet IEEE Int. Conf. Netw. Educ. Res. RoEduNet 2018*, pp. 1–6, 2018.
- [13] A. Mardiyono, W. Sholihah, and F. Hakim, "Mobile-based Network Monitoring System Using Zabbix and Telegram," *2020 3rd Int. Conf. Comput. Informatics Eng. IC2IE 2020*, pp. 473–477, 2020.
- [14] D. Arnaldy and T. S. Hati, "Performance Analysis of Reverse Proxy and Web Application Firewall with Telegram Bot as Attack Notification on Web Server," *2020 3rd Int. Conf. Comput. Informatics Eng. IC2IE 2020*, pp. 455–459, 2020.
- [15] S. Sulasno and R. Saleh, "Desain dan Implementasi Sistem Monitoring Sumber Daya Server Menggunakan Zabbix 4.0," *JUITA J. Inform.*, vol. 8, no. 2, pp. 187, 2020.
- [16] Cisco Press, "Cisco 's PPDIIO Network Cycle," 2011. [Online]. Available: <https://www.ciscopress.com/articles/article.asp?p=1608131&seqNum=3>. [Accessed: 15-Feb-2021].
- [17] S. P. Deriyanto and H. A. Santoso, "Development of Bot for Microservices Server Monitoring Using Life Cycle Approach to Network Design Method," *JUITA J. Inform.*, vol. 8, no. 2, pp. 141, 2020.
- [18] L. Hernandez, G. Jimenez, A. Pranolo, and C. U. Rios, "Comparative Performance Analysis Between Software-Defined Networks and Conventional IP Networks," *Proceeding - 2019 5th Int. Conf. Sci. Inf. Technol. Embrac. Ind. 4.0 Towar. Innov. Cyber Phys. Syst. ICSITech 2019*, pp. 235–240, 2019.
- [19] B. Mayer and R. Weinreich, "A dashboard for microservice monitoring and management," *Proc. - 2017 IEEE Int. Conf. Softw. Archit. Work. ICSAW 2017 Side Track Proc.*, vol. 6, pp. 66–69, 2017.
- [20] J. Renita and N. E. Elizabeth, "Network's server monitoring and analysis using Nagios," *Proc. 2017 Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2017*, pp. 1904–1909, 2018.
- [21] P. Jain, Y. Munjal, J. Gera, and P. Gupta, "Performance Analysis of Various Server Hosting Techniques," *Procedia Comput. Sci.*, vol. 173, pp. 70–77, 2020.
- [22] K. Rafiullah, S. Ullah, "Design and implementation of an automated network monitoring and reporting back system," *Journal of Industrial Information Integration*, vol. 9, pp. 24–34, 2018.

