# JESTEC_Banu Santoso_Turnitin

*by* Banu Santoso

**Submission date:** 02-Dec-2020 07:02AM (UTC+0700)
**Submission ID:** 1461899508
**File name:** JESTEC_Banu_Santoso.docx (153.94K)
**Word count:** 3437
**Character count:** 18320

6
Journal of Engineering Science and Technology
Vol. XX, No. Y (Year) PPP - QQQ


# IMPROVEMENT OF SETUP TIME ON SERVER INFRASTRUCTURE AUTOMATION USING ANSIBLE FRAMEWORK

**Abstract**

System administrators must be prepared to quickly and accurately design the server infrastructure, due to the development of technology in the server field this day. A way to manage servers is to use the Ansible framework, which can be used to install and configure multiple servers at once. Ansible able to make installation and configuration on multiple servers easier than conventional methods (shell scripts), and future configurations can be easily adjusted centrally. This study aims to manage servers automatically and measure setup time using the Ansible framework. The method used is to access Alibaba Cloud, then the sysadmin creates access_key & secret_key on the profile page. Once created the Ansible script can be run using a terminal or cmd and receive server information such as Hostname, IP Ad-dress and Web Address that can be accessed. The results of this study indicate that the automation method using Ansible can shorten the time of the entire installation and configuration. The measurement results show that if using the conventional way, the average time needed for each server is 27 minutes 28 seconds and using Ansible, the average time it takes is 3 minutes 31 seconds.

Keywords: Ansible, Server, Infrastructure, Automation, Shell Script

## 1. Introduction

Technology in the server field is experiencing very rapid growth [1] and allows an administrator to be able to handle more than one server [2-4], causing an administrator to experience difficulties in server management [5-6]. Therefore, it is necessary to have a DevOps approach as a bridge between the developer (Dev) and the Ops (Sys-tem) to make it easier to handle quickly and in control of the system to be developed [7-9]. The in-house server infrastructure for website creation carried out by DevOps still uses the traditional way, from buying a server to configuring servers for production, testing, and development that is done repeatedly [10-11]. The pro-cessing time for this server infrastructure to be faster and more efficient is what the company hopes for [12]. So that we need a method so that the creation and configuration of servers can be done quickly, precisely, and easily [13].

Based on the aforementioned problems, it takes a method of making infrastructure that is generated automatically using an Ansible application [14]. In the future, this method is expected to be used with the aim that the person in charge of the server infrastructure for the provision of the need for developers to start a project [15-17]. Through the Ansible framework, one script can be created by DevOps, then it can be run directly to many servers [18]. In addition, the centralized configuration makes Ansible easy to change according to the needs of any website application quickly [19-20].

The Ansible framework which is used for the automation process can significantly speed up the server configuration process and is on target, and will also simplify the multi-server configuration simultaneously [21]. The objective of this research is to improve setup time of server infrastructure automation using Ansible Framework.

## 2. Literature Review

### 2.1. State of The Art

The contribution of research results related to reducing setup time in cloud services, such as deploying server infrastructure, building server infrastructure, and building applications using automation that has been done by several previous studies, is presented in Table 1. The table contains studies - research in terms of researchers, location, research contribution, and methods.

**Table 1. Research Overview**

| Researcher/Year | Location | Research Contribution | Methods |
|---|---|---|---|
| **Yasuharu Katsuno, dkk / 2015 [22]** | Virtual Machine, IaaS | Reduces time on application deployment across IaaS services and Instances | Chef |
| **Nishant Kumar Singh, dkk / 2015 [23]** | Amazon AWS | Time savings in provisioning servers on AWS EC2 | Ansible, AWS API |
| **Markus [illegible]opperi / 2017 [24]** | Amazon AWS | Time efficiency and monitoring in CICD | ChatOps dengan |

| | | | |
|---|---|---|---|
| **Somya Garg, dkk / 2019 [25]** | Google Cloud | Reduces time on Cloud Infra-structure, and CICD | Hubot, Ansible Docker dengan Robust Container Security |
| **K. Sree Poornalinga/ 2016 [26]** | Amazon AWS | Reduces time on the CIDD infrastructure | Jenkins, Maven |
| **This research** | Alibaba Cloud | Reduced time in server infrastructure deployment and configuration on Alibaba Cloud | Ansible, Alibaba Cloud API |

Based on the literature study there are differences in methods and can be proposed as a positive contribution to this research. This is in the form of designing and creating automation scripts using Ansible and Alibaba Cloud Api to improve the processing time of the server infrastructure and facilitate the rapid deployment of applications that are being developed to the server [25].

Research conducted by Yasuharu Katsuno et al. [22] which explains that the deployment time can be de-creased by up to 40% with the parallel approach method on virtual machines using the Chef framework. The advantage of using Chef is that it can significantly reduce the spreading time of fish. The weakness of this research is that the implementation is not very User Friendly, which makes the time to learn this method very long.

Research conducted by Nishant Kumar Singh et al. [23] discussing Automated Provisioning of Applications in the IAAS Cloud using Ansible Configuration Management, said that tools can now help Devops build servers on AWS EC2 in less than a minute compared to using the method. it can take a week to configure Hardware on AWS EC2. Automation can reduce infrastructure provisioning time from days to hours. The advantage of this research is that it can reduce the time to build servers on AWS EC2 and help Devops build server infrastructure. The drawback of this research is that DevOps must optimize the automation process before it can run automation, because running automation in an un-optimized state can interfere with productivity.

Moreover, Markus Juopperi [24] conducted research on the deployment of applications to servers using Chatbots, namely ChatOps and Ansible. The method used in this study can reduce time and efficiency in the deployment of applications. The advantage of this study that uses ChatOps is visibility. Every action taken is seen by everyone on the team. Action history is also saved in chats. It provides a clear histo-ry of what has been deployed for production or is in testing across the team.

In addition, Somya Garg et al. [25] conducted research to reduce the time for continuous Integra-tion and Sustainable Shipping or commonly known as CICD using Docker with Ro-bust Container Security. This research also provides a strategy for the deployment and maintenance of automated resources so that they

can be managed effectively. The advantage of this research is that the use of virtualization can reduce the time required to use these computing resources from days to minutes. The drawback of this research is that Docker requires large server resources, so it can increase the cost of renting a server with large resources.

Furthermore, Sree Poornalinga [26] discussed about an effective automation system that can help save time and costs in software quality and productivity. This research describes automation in application builder from source code using Maven and Jen-kins. Through this concept can reduce waiting time and find errors from the source code because Jenkins will check the source code and notify the developer that there is an error in the source code that Jenkins built. The advantage of this research is that the application building process can be completed in a short time. The drawback of this research is the lack of knowledge about sustainable integration tools and their environment.

## 2.2. Ansible

Ansible is an open-source automation tool for managing and configuring computers. Ansible is developed by Red Hat and the open source community. Ansible is designed to manage complex infrastructure rather than a single case. Dynamic inventory of instances, which is required when instances come and go, is supported with additional packages for common cloud providers.

The method used in this study is the Ansible method as presented in Table 2. The reason for choosing to use the Ansible method compared to other methods is that it has advantages including that it is easier in the setup process, Ansible management is also easy, then it is cheap to use up to 100 nodes. by using Ansible Tower, or free of charge by using the AWX Cloud. In making the Ansible script it is also easy to understand by System Administrators and DevOps because it uses the YAML Configuration File that is easy to understand and learn and administrator-oriented. Ansible distribution uses SSH, so that the server to be setup does not require additional setup and the server setup process is faster.

**Table 2. Comparison of Automation Software** [27] [28]

| Classification | Chef | Puppet | Ansible | Saltstack |
|---|---|---|---|---|
| Availability | ✓ | ✓ | ✓ | ✓ |
| Setup Easiness | Difficult | Difficult | Easy | Difficult |
| Management | Difficult | Difficult | Easy | Easy |
| Scalability | Very measurable | Very measurable | Very measurable | Very measurable |
| Configuration Language | DSL (Ruby) | DSL (PuppetDSL) | YAML (Python) | YAML (Python) |
| Interoperability | High | High | High | High |
| Price (up to 100 node) | $13700 | $11200-$19900 | $10,000 | $15,000 (Estimation) |

## 3. Research Methods

### 3.1 Materials

There are several tools and materials in the form of hardware and software used in this research. The materials and tools used will be described in the following subsections.

#### 3.1.1 Hardware

The hardware used in this study is as follows.
a. CPU 1.6 GHz Intel Core i5-8250U
b. Memori DDR 16 GB
c. NVIDIA GeForce 940MX
d. Linux Ubuntu Operating System version 16.04 LTS and Windows 10
e. SSD storage with 250 GB capacity

#### 3.1.2 Software

The software used in this study are as follows.

a. Alibaba account to get access_key and secret_key and server instances.
b. Nginx web server to handle requests from the internet which are processed by the backend and respond back to the internet.
c. PHP programming language to process requests requested by the Nginx web server and forwarded to the PostgreeSQL database. After that it will be for-warded back to the Nginx web server.
d. Composer is a PHP-specific dependency manager that has functionality like Gem (Ruby) or Maven (Java). The composer library will automatically in-stall the other libraries needed without the need to download one by one.
e. The PostgreSQL database is a robust open source object relational database system. The PostgreSQL database has over 15 years of active development and a proven architecture that has earned strong reputation for reliability, data integrity, and correctness.

### 3.2 Research Flow

Figure 1 is a flowchart performed by the system administrator if using Ansible. In Figure 1 it can be explained, before setting up server automation, you must have an account from the selected cloud, here using Alibaba Cloud. If you already have an Alibaba Cloud account, the sysadmin creates access_key & secret_key on the profile page. This key is used to access, create, delete services from Alibaba Cloud. After the sysadmin gets the access_key and secret_key, the sysadmin can generate an Ansible script for automated server creation. Once created the Ansible script can be run using a terminal or cmd and receive server information such as Hostname, IP Ad-dress and Web Address that can be accessed.

The IP address obtained from creating the server is used to automate server configuration, such as software installation, configuration and fetching project data on GitHub, GitLab and so on.

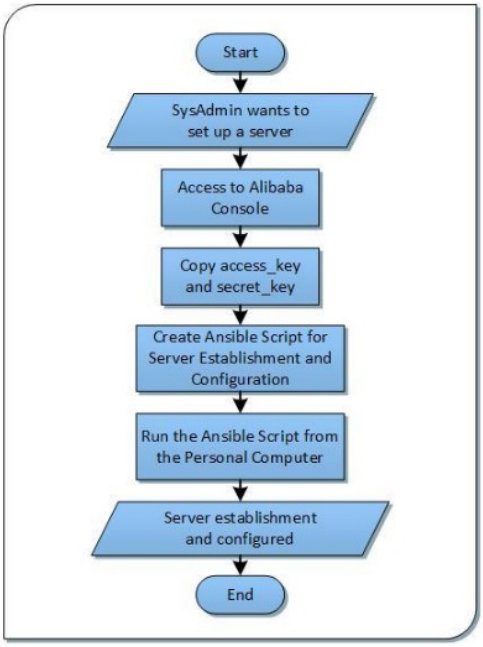


**Fig. 1. Research Flowchart**

### 3.3 System Design

This system design block diagram can be seen in Figure 2. The system for improving setup and configuration times in automating server infrastructure using the Ansible method.

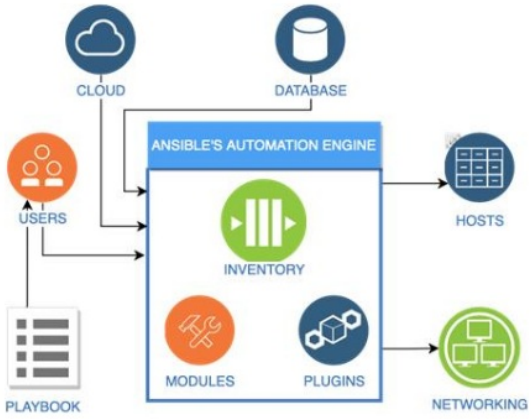


**Fig. 2. System Architecture**

Things that need to be prepared when automating the Cloud server include Server Accounts, namely Access_Key and Secret_Key. After obtaining Access_Key and Secret_Key, the System Administrator can develop Script Ansible to automate the Server Infrastructure. The Ansible script can be run using a terminal command prompt to get server information in the form of an IPv4 or IPv6 Address, Hostname and Web Address (Domain Name Server). The IP address information on the server is used to automate server configuration including the installation of required software, configuration and data retrieval on Gitlab, Github and others. The Playbook consists of several collections of roles with specific tasks for working on modules that will be installed and configured. The Module is the default application that is used to install or prepare several applications. Plugins is an extension from Ansible, so that Ansible can run as desired. Whereas in the Inventory is a collection of lists of addresses or IP address servers that will be directed to be configured automatically on the server.

## 4. Results and Discussion

### 4.1. Comparison of Processing Time and Hardware Specifications

Testing is done by testing the script that has been made on the server and testing using the traditional method. The test results on the script and traditional include measuring the time of Setup to servers that are made with different specifications, for testing the script starts when DevOps runs the script until the resulting output is that the server is installed and configuration can display the web application, while for traditional it starts when DevOps enters the server using SSH until the resulting output, namely the configured server, can display web applications. The results of the server setup time measurement are presented in Figure 3 and Figure 4.

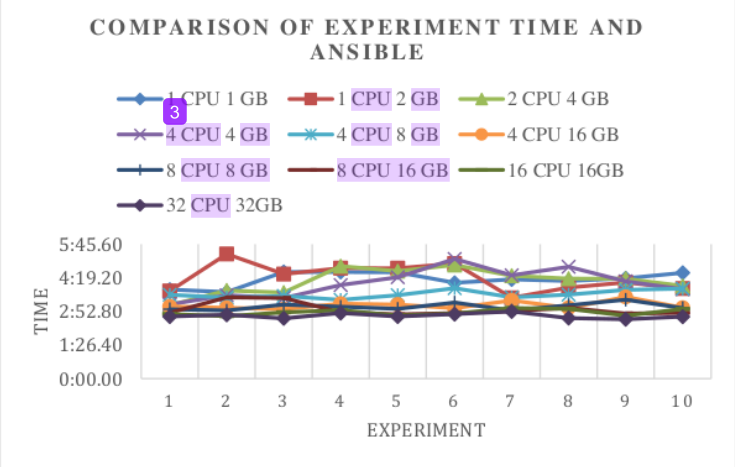


**Fig. 3. Comparison of Experiment Time and Ansible**

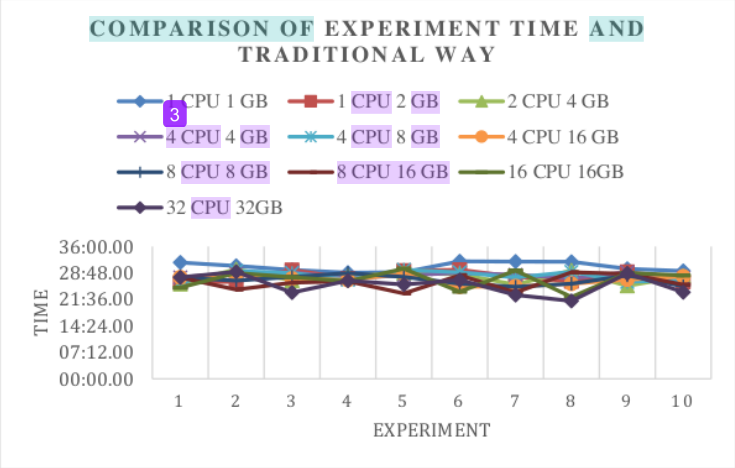


**Fig. 4. Comparison of Experiment Time and Traditional Way**

The average setup time using the traditional method is 25 minutes 34 seconds to 30 minutes 29 seconds, while using Ansible is 2 minutes 41 seconds to 4 minutes 15 seconds. From 10 experiments, it can be taken the average, longest time, fastest time and standard deviation which are presented in Table 3.

**Table 3. Comparison of the setup time of 10 trials**

| Information | Ansible | Traditional |
|---|---|---|
| **Average** | 03:31,35 | 27:28,82 |
| **Fastest Time** | 03:04,50 | 25:03,80 |
| **Longest Time** | 04:00,36 | 29:33,71 |
| **Deviation Standard** | 00:18,46 | 01:33,03 |

### 4.2. Server Setup Time

The comparison chart of server setup time between Ansible and traditional is presented in Figure 5. In this graphic image, the traditional setup time span is above 21 minutes and under 40 minutes, while the time span for Setup using Ansible is above 2 minutes and under 6 minutes.

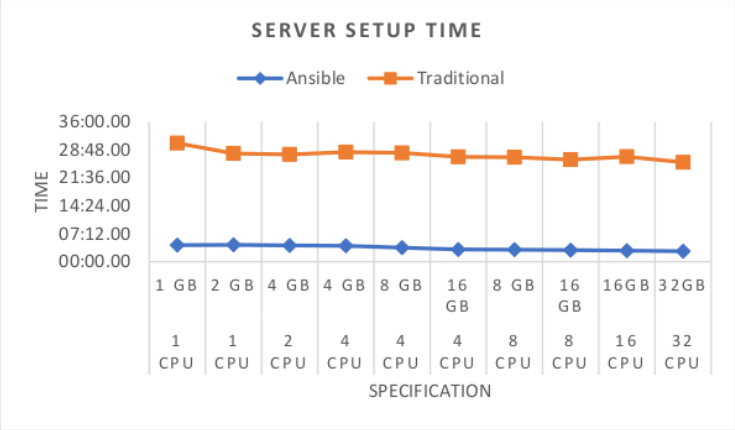


**Fig. 5. Server Setup Time**

In testing the setup time using the traditional method there are problems when configuring, so even though using high server hardware, there is no significant reduction in time. Ansible, meanwhile, shows the decreasing time on different server hardware.

## 5. Conclusions

Based on the formulation of the problem about the length of time to create a VPS server setup on Alibaba Cloud and the results of the research that has been done, it can be concluded that making a server on Alibaba through the console from the Alibaba website takes a long time, an average of 27 minutes. In this study, a solution is offered using an automation method through applications such as Ansible to make servers on Alibaba can be made in one script with a relatively faster time with an average time of 3 minutes 30 seconds. Automation methods using Ansible can significantly increase installation and configuration times on multiple servers. In this study, trials were carried out and experienced a reduction in time of more than 5 hours in the traditional way. By way of Ansible it took less than 1 hour. The test results show that the higher the server specifications on Alibaba are used, the faster the setup process using the Ansible method will be. For future work, it can discuss about the use of Ansible to automate the configuration process on the server and can be used to configure the server for load balancing. In addition, the Ansible Script to automate server creation on Alibaba can be made to be adapted to other Cloud Providers, for example Amazon Web Service or Microsoft Azure.

## References

1. Kheirabadi, C.; and Groulx, D. (2016). Cooling of server electronics: A design review of existing technology. *Appl. Therm. Eng.*, vol. 105, no. 2016, pp. 622–638.
2. Inam ul haq; Wang, J.; Zhu, Y.; and Maqbool, S. (2020). An efficient hash-

based authenticated key agreement scheme for multi-server architecture resilient to key compromise impersonation. *Digit. Commun. Networks*, no. May.

3. Liu, H.; Bao, C.; Xie, T.; Gao, S.; Song, X.; and Wang, W. (2019). Research on the intelligent diagnosis method of the server based on thermal image technology. *Infrared Phys. Technol.*, vol. 96, no. 61402074, pp. 390–396.
4. Padhy, N. (2020). An automation API to optimize the rate of transmission using rclone from local system to cloud storage environment. *Mater. Today Proc.*, no. xxxx.
5. Ali, E.; Susandri; and Rahmaddeni. (2015). Optimizing Server Resource by Using Virtualization Technology. *Procedia Comput. Sci.*, vol. 59, no. Iccsci, pp. 320–325.
6. Liu, H.; Bao, C.; Xie, T.; Gao, S.; Song, X.; and Wang, W. (2019). Research on the intelligent diagnosis method of the server based on thermal image technology. *Infrared Phys. Technol.*, vol. 96, no. 61402074, pp. 390–396.
7. Lassenius, C.; Dings, T.; and Paasivaara, M. (2015). DevOps: A Definition and Perceived Adoption Impediments. *Lect. Notes Bus. Inf. Process.*, vol. 212, pp. 166–177.
8. Babar, Z.; Lapouchnian, A.; and Yu, E. (2015). Modeling DevOps Deployment Choices Using Process, vol. 2, no. June 2016, pp. 322–337.
9. Lwakatare, L.E.; Kilamo, T.; Karvonen, T.; Sauvola, T.; Heikkilä, V.; Itkonen, J.; Kuvaja, P.; Mikkonen, T.; Oivo, M.; and Lassenius, C. (2019). DevOps in practice: A multiple case study of five companies. *Inf. Softw. Technol.*, vol. 114, no. March 2017, pp. 217–230.
10. Forti, S.; Ferrari, G.L.; and Brogi, A. (2020). Secure Cloud-Edge Deployments, with Trust. *Futur. Gener. Comput. Syst.*, vol. 102, pp. 775–788.
11. Arcangeli, J.P.; Boujbel, R.; and Leriche, S. (2015). Automatic deployment of distributed software systems: Definitions and state of the art," *J. Syst. Softw.*, vol. 103, pp. 198–218.
12. Lloyd, W.; Pallickara, S.; David, O.; Lyon, J.; Arabi, M.; and Rojas, K. (2013). Performance implications of multi-tier application deployments on Infrastructure-as-a-Service clouds: Towards performance modeling. *Futur. Gener. Comput. Syst.*, vol. 29, no. 5, pp. 1254–1264.
13. Tso, F.P.; Jouet, S.; and Pezaros, D.P. (2016). Network and server resource management strategies for data centre infrastructures: A survey. *Comput. Networks*, vol. 106, pp. 209–225.
14. Masek, P.; Stusek, M.; Krejci, J.; Zeman, K.; Pokorny, J.; and Kudlacek, M. (2018). Unleashing full potential of ansible framework: University labs administration. *Conf. Open Innov. Assoc. Fruct*, vol. 2018-May, pp. 144–150.
15. Dalla, S.; Di, D.; and Tamburri, D.A. (2020). SoftwareX AnsibleMetrics : A Python library for measuring Infrastructure-as-Code blueprints in Ansible. *SoftwareX*, vol. 12, p. 100633.
16. Pribiš, R.; Beňo, L.; and Drahoš, P. (2019). Implementation of Micro embedded OPC Unified Architecture server-client. *IFAC-PapersOnLine*, vol. 52, no. 27, pp. 114–120.
17. J. Martinez, J;, Dasari, D.; Hamann, A.; Sañudo, I.; and Bertogna, M. (2020). Exact response time analysis of fixed priority systems based on sporadic servers. *J. Syst. Archit.*, vol. 110.
18. Mazumdar, S.; and Pranzo, M. (2017). Power efficient server consolidation for Cloud data center. *Futur. Gener. Comput. Syst.*, vol. 70, pp. 4–16.

19. Kumar, R.; and Goyal, R. (2020). Modeling continuous security: A conceptual model for automated DevSecOps using open-source software over cloud (ADOC). *Comput. Secur.*, vol. 97, p. 101967.
20. Keupondjo, G.A.S.; Anoh, N.G.; Adepo, J.C.; and Oumtanaga, S. (2019). Hybrid routing with latency optimization in SDN networks. *J. Eng. Sci. Technol.*, vol. 14, no. 5, pp. 3062–3072.
21. Petrochina, W.Y.; Petrochina, Z.T.; and Petrochina, G.Y. (2018). Design and implementation of continuous integration scheme based on Jenkins and Ansible. *2018 Int. Conf. Artif. Intell. Big Data, ICAIBD 2018*, pp. 245–249.
22. Katsuno, Y.; and Takahashi, H. (2015). An automated parallel approach for rapid deployment of composite application servers. *Proc. - 2015 IEEE Int. Conf. Cloud Eng. IC2E 2015*, pp. 126–134.
23. Singh, N.K.; Thakur, S.; Chaurasiya, H.; and Nagdev, H. (2016). Automated provisioning of application in IAAS cloud using Ansible configuration management. *Proc. 2015 1st Int. Conf. Next Gener. Comput. Technol. NGCT 2015*, no. September, pp. 81–85.
24. Juopperi, M. (2017). Deployment automation with ChatOps and Ansible, p. 22, 2017.
25. Garg, S.; and Garg, S. (2019). Automated Cloud Infrastructure, Continuous Integration and Continuous Delivery using Docker with Robust Container Security. *Proc. - 2nd Int. Conf. Multimed. Inf. Process. Retrieval, MIPR 2019*, pp. 467–470.
26. Poornalinga, K.S.; and Rajkumar, P. (2016). Continuous Integration, Deployment and Delivery Automation in AWS Cloud Infrastructure. *Int. Res. J. Eng. Technol.*
27. Zhang, G.; and Ravishankar, M.N. (2019). Exploring vendor capabilities in the cloud environment: A case study of Alibaba Cloud Computing. *Inf. Manag.*, vol. 56, no. 3, pp. 343–355.
28. Hochgeschwender, N.; Biggs, G.; and Voos, H. (2018). A reference architecture for deploying component-based robot software and comparison with existing tools. *Proc. - 2nd IEEE Int. Conf. Robot. Comput. IRC 2018*, vol. 2018-Janua, pp. 121–128.

# JESTEC_Banu Santoso_Turnitin

ORIGINALITY REPORT

| 8% | 8% | 1% | 6% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | www.theseus.fi<br>Internet Source | 3% |
| 2 | www.prototypingtool.com<br>Internet Source | 1% |
| 3 | www.deisa.eu<br>Internet Source | 1% |
| 4 | Submitted to Taylor's Education Group<br>Student Paper | 1% |
| 5 | jestec.taylors.edu.my<br>Internet Source | 1% |
| 6 | mafiadoc.com<br>Internet Source | 1% |
| 7 | "Generator Performance in Steam Power Plant", International Journal of Innovative Technology and Exploring Engineering, 2019<br>Publication | 1% |

| | | | |
|---|---|---|---|
| Exclude quotes | On | Exclude matches | < 1% |
| Exclude bibliography | On | | |