

# ANALYSIS OF IMPLEMENTATION CONTAINER-BASED MICROSERVICES ARCHITECTURES WITHIN THE OPEN SOURCE SOFTWARE DEVELOPER COMMUNITY (CASE : OPENDAYLIGHT DEVOPS COMMUNITY)

AHMAD NURUL FAJAR<sup>1</sup>, RAHMAD ADE PUTRA<sup>2</sup>

<sup>1,2</sup> Information Systems Management Department, BINUS Graduate Program-Master of Information Systems

Management, Bina Nusantara University Jakarta, Indonesia 11480

E-mail: <sup>1</sup>afajar@binus.edu, <sup>2</sup>rahmad.a.pt@binus.ac.id

## ABSTRACT

In the research, we aim to be able to examine the success factors of the implementation of microservices container-based architecture in the scope of Open source project called as Opendaylight community. These factors which have major influences to the implementations of microservices containers are source code commit, merge source code, installed dependency, connection latency to master node, construction time. The results of test tests on published data for implementing microservices architecture show that using a regression model can find the level of influence of variables that exist during the implementation process as well as solve the issue. The factors that most influence the level of success of the implementation according to the trials carried out are merge source code, installed dependency, connection latency and construction time

**Keywords:** *Microservices Artitecture, Opendaylight, Container*

## 1. INTRODUCTION

One of the hottest topics in Information and Communication Technology today is cloud computing or that is commonly referred to as the cloud computing host. As a trend and buzzword on information technology and telecommunications that are still being developed by ICT practitioners to date [2]. Cloud computing technology is presented as a form of the business to enable access to resources and applications from anywhere through computer networks either internet and intranet, so that case the limitations and shortcomings of the use of ICT infrastructure that previously existed can be overcome and tackled. In this study we will discuss the analysis and assessment of one of areas that includes cloud computing technology, namely container-based private cloud computing. Container technology is a technological concept which can isolate a process from another process by breaking it into libraries and application dependencies or supporting software packages used in operating systems without using the entire set of operating systems as

monolithic, or in other words container can be called as a shared operating systems [6]. The main software in container technology usually uses Docker on the Linux operating system and uses Kubernetes as an orchestration or management container that allows to deploy or implement software application projects into a variety of isolated virtual isolation environments at the operating system level [1]. OpenDaylight is a collaborative open source project development in the Linux Foundation with Apache 2.0 licensed which aims to accelerate the adoption and application of Software-Defined Networking (SDN) in the enterprise environment. The Opendaylight developer community uses Bitergia Analytics to display all the statistics of the application development process and the process of making alternative lighting components based on the CI / CD automation process on Jenkins. At the writing of this study, we will take the data that is on Jenkins, where the data is in the form of statistical automation of source code that is periodically and continuously built by developers or contributors of developers to microservices containers. The data used as a source for analysis is data that exists

within 1 year from October 1, 2017 to October 1, 2018. The data is in the form of CSV or Comma-separated Values files that are constructed from data mining in statistical contributors to the development of the Microservices component. At the level of implementation of microservices in the development environment Opendaylight is only limited to the development environment, so there is no de-facto standard yet to determine the use of container, therefore the analysis phase is needed to determine the feasibility factor for microservices architecture [1]. Factors that influence the level of success and also the failure factor in the implementation process of microservices in the development environment of the Openlight project is Average Source Code Commit in Git which is the number of commit averages or which means changes to the source code before sending to the master repository to Git. Average Merge Source Code is the average number of merging between branches of the source code in git to become a standard into an image container in the form of a master branch. Number of Installed Dependency Packages are the sum of the number value of the dependency package, either the module or library that is on the microservices container and encapsulated into a package of image containers that goes through the Jenkins automation process. Number of connection latency to master node is the average number of latencies in the Microservices container when it is in the process or state of build, test and deploy, resulting in a container package unit processed by Jenkins master server. Average construction time is the average amount of time needed when the cron process or automation takes place, the process is in the form of build, test and deploy. Total container node on Container is the number of all major application code sources that are available and implemented on microservices containers, in this case the total number of source code is calculated based on integration of git plugins with Jenkins master nodes in the period October 1, 2017 to October 1, 2018.

## 2. RELATED WORKS

### A. *Microservices Architecture*

Microservice is a collection of independent and small processes which basically can communicate with each other to form the complex applications that are agnostic regards of any API language [6]. These services consist of small, separate blocks and focus on light tasks to facilitate modular methods in system development. Microservice itself is an advanced development of Service-oriented

architecture (SOA) because Microservice is a system consisting of component components in the form of modular, autonomous services that have their respective goals but are orchestrated through light-weight protocols with each other to achieve a single goal certain especially in software development [1].

### B. *Jenkins*

Jenkins is an open source automation server to automate tasks in the process of continuous integration and delivery of software. Jenkins is a Java-based application that can be installed from the Ubuntu repository or by downloading and running Web applicatino ARchive (WAR) files, a collection of files that are complete and just run on a server [4].

### C. *Jenkins Build*

Jenkins' build is a feature that can run and build a job to automate controlled and monitored source code to become a Docker image, every activity that is in it. One example of a build or development of automation of the source code to be built into an image container is compiling code sources, running tests, implementing module packages and libraries [4].

### D. *Jenkins Pipeline*

The pipeline in Jenkins is a series of plugins that have support for the development and application integration process with continuous delivery. In this case the existing pipeline in CI / CD is an automation process from the process of getting an application version developed based on managing source code such as Git or Subversion.

### E. *Bittergia Analytics*

It is a web-based application for analyzing the software development process as a whole project. In Bittergia Analytics provides statistical data based on final data and insights to be displayed to software developers who collaborate with each other. The use of Bittergia Analytics is usually at the corporate level or open software community that wants to openly display data about processes, reports and data analysis.

### F. *Kibana*

It is a web-based application for analyzing the software development process as a whole project. In Bittergia Analytics provides statistical data based on final data and insights to be displayed to software developers who collaborate with each other. The use of Bittergia Analytics is usually at

the corporate level or open software community that wants to openly display data about processes, reports and data analysis.

#### G. Opendaylight Project

OpenDaylight Project is a collaborative open source project at the Linux Foundation that aims to accelerate the adoption of Software-Defined Networking (SDN) and create a solid foundation for Network Function Virtualization (NFV) with a more transparent approach to encourage new innovations and reduce risks. The Linux Foundation Collaborative Projects is an independently funded software project that aims to harness the power of collaborative development to fuel innovation in industry and ecosystems [5].

#### H. Docker Container

Docker is an open source project aimed at software engineers, DevOps engineers or sysadmins to build, package and run applications anywhere in a container. Docker functions as virtualization of an operating system or a server or a web server or even a database server, where by using this virtualization, it is expected that developers can develop applications according to server specifications.

#### I. Git

Git is a version control system that is used by developers to develop software together. The main function of git is to set the version of your program source code by giving a line mark and which code is added or deleted or even modified and merging between branches or code projects [4].

#### J. JASP

JASP is a graphic and open-source program for statistical analysis that is designed to be easy to use just like SPSS. JASP provides many Bayesian statistical methods and can produce tables and plots that are easy to understand. The JASP development is supported financially by several universities and research funds. JASP offers conclusions according to a statistical model that uses p values and confidence intervals to control infinite error levels in infinite replication limits so that it can estimate credible parameter values and proof the model given data according to existing knowledge.

### 3. RESEARCH METHOD

In doing this research, we use a research approach by applying the descriptive approach method and the verification approach method, because there are variables which will be related to each other and the

relationship between one and the other, and the purpose to present a structured, accurate picture of the grouping of variables and relationship between the variables studied, namely the effect of success on the implementation process on the development of ODL.

The purpose of the descriptive study was to find out and analyze the influence of the independent variables on Bitergia Analytics Opendaylight on factors that influence the level of success in implementing Opendaylight. Verification research methods basically want to test the truth of a hypothesis that is implemented through data collection in the community.

#### K. Research Object

The object of research is a reference to data sources that become benchmarks and also attention in a series of studies, while research objects are targeted in research, namely to get answers or solutions to problems that are occurring, which solutions are useful in the future after the research is completed and can useful to develop.

#### L. Research Model

This research is applied research that aims to apply, test, and evaluate data analysis capabilities applied in solving practical problems with an experimental approach to the open source collaboration software development project community. The experimental approach was chosen to look for the influence of each independent variable on the dependent variable under tightly controlled and rigorous conditions. Regarding the level of exploration, this study uses explanatory research to test certain hypotheses in order to justify or reinforce hypotheses, look for causes of a symptom and determine the nature of the relationship between one or more symptoms or dependent variables with one or more independent variables. Besides that, this study uses correlational design, which can aim to determine the relationship of the independent variable with the dependent variable.

So that conclusions can be drawn about validation for the use and standardization of Microservices architecture which aims to simplify the consumption of resources in the production environment in the enterprise so that it can save resources.

#### M. Collecting Data

The variables described in this study are clearly defined so that they do not lead to a double





		91.47	15.		9.3	1.2
didm	37	3	46	33	8	2
distribut	10,5	1,216	23.	4,10		19.
ion	91	.93	02	9	0	35
		35.67	17.		5.9	11.
dlux	256	1	98	106	1	82
dluxapp		174.6	19.		6.6	12.
s	385	52	56	272	8	26

the form of a master branch. Average Installed Dependency Packages are the sum of the average value of the dependency package, either the module or library that is on the microservices container and encapsulated into a package image container that passes the CI / CD pipeline automation process from the Jenkins server Opendaylight project, this variable can be classified as independent ( X3). Average Connection Latency to Master Node is the average number of letencies that exist in Micorservices containers while in the process of build, test and deploy, resulting in a container package unit processed by Jenkins master server, this variable can be classified as independent variable or (X4) . Average Construction Time is the average amount of time needed when the cron process or automation takes place, the process is in the form of build, test and deploy, in this variable can be classified into independent variables (X5).

*N. Analysis Data Method*

The data analysis method used in this study is to use quantitative analysis on the dataset obtained. Data analysis can be done using multiple linear regression tests consisting of t statistical tests and F statistical tests to prove the hypotheses formed in the study and to see the level of significance.

Regression analysis is basically an approach used to define mathematical relationships that connect between the dependent variable (Y) with one or several independent variables (X). Mathematical relationships are used as a regression model that is used to predict or predict the output value (Y) based on a particular input value (X). With the regression analysis method will be known independent variables that really can significantly affect between one variable with another so that it can be drawn as well as evidence of comparative data. Before the model test, we conducted descriptive statistical analysis, as well as the classic assumption test of the regression model which included normality test, autocorrelation test, heterocedasticity test, multicollinearity test, and linearity test.

As a manner to minimize errors or human error in order to processing statistical data and absolute research data, we use JASP Statistics programs or applications. The JASP program is an Open Source licensed application program which is open software that is able to analyze statistics in full. This program was chosen by us because it has several advantages over other programs or software, namely this program is user-interface based on many systems, especially windows, and the program is very user-friendly and complete statistical analysis techniques are available. This

Based on the table 1 above, this study uses a large sample microservices container data sample of 80 container nodes and cannot be displayed as a whole on the attachment page or in this writing. Data is minimized so that only 20 samples can be displayed or in other words N1 to N-20, after that the table towards the 80th sample line will be entered into the index table at the end of the page. From all the data above based on Jenkins master's statistics, there is the Microservice Container Images, which are container names in microservices in the form of sample population data and also dependent data or (Y), while success factors in CI / CD pipeline implementation on each microservices container sample are written with Success Node Creation as part of the dependent variable (Y). The secondary data is the result of visualization of Bitergia Analytics by using Kibana as a software development tracking progress or an application that can monitor all software development collaborations between developers in organizations or communities, in this case the Opendaylight Project microservices container tracking.

Based on the dependent variable or (Y) above, then the next is data statistics with Independent variables in the continuous integration process and continuous delivery pipeline. Average Source Code Commit or Trigger on Commit on the Jenkins server master is a number of independent variables or (X1) which is the number of commit averages or which means changes to the source code that exists before being sent to the master repository to Git. Average Merge Source Code or trigger on merge is classified as an independent variable (X2) on Jenkins server. It is the average number of merging between branch source code in git to become a standard into an image container in



program has been commonly used by previous research to conduct the analyzed of the research data. Various scientific disciplines, both the scope of management (marketing research), biology, agriculture, engineering, industry, psychology, social sciences and other fields, use this software as a tool to process / analyze research data.



Figure 1. JASP GUI

#### O. Statistical Descriptive Data Analysis

This statistical analysis is used to analyze data by describing or describing data that has been collected as it is without intending to make conclusions that apply to general or generalizations. Preliminary data analysis is conducted to classify, sort and simplify data so that it is easy to read and interpret. This general description can be a reference to see the characteristics of the data that can be obtained. This is very important because with descriptive analysis it is possible to be able to quickly correct data that has been sent. Descriptive statistical analysis used in this study consisted of tabulations and percentages. In this case, the data in microservices container names are categorized based on microservices pod containers, which are container groups based on the type of operating system and modules and existing libraries.

The frequency distribution tabulation is made for nominal data which shows the magnitude of each independent variable in the sample. The percentage shows the proportion of the independent variables in the sample. Frequency distribution tables are useful for describing the characteristics or characteristics of a variable, studying the distribution of the main variables and choosing the classification of the principal classification for cross tabulation.

- Minimum, maximum, mean and standard deviation

The data will be used to indicate the level of the dependent variable. The minimum shows the lowest value of the data in the sample, whereas maximum shows the highest value of the data in the sample. The mean is used to determine the average data in the sample. The smaller the standard deviation of a variable in this study means the smaller the distribution, which means the value of the data is more homogeneous. Conversely if the greater the distribution means the more varied the value of the data.

#### P. Multiple Linear Regression Tests

The statistical test used is by using multiple linear regression statistical tests. This analysis is used to calculate and obtain an idea of how the independent variables (X) and the dependent variable (Y) influence.

#### Q. Double Correlation Test and Determination Coefficient Test

The coefficient of determination ( $R^2$ ) measures how far the ability of the model to explain the variation of the dependent variable. Between zero and one. The small value of  $R^2$  means that the ability of independent variables to explain variations in the dependent variable is very limited. A value close to one means that the independent variables are able to provide almost all the information needed to predict the variation of the related dependent variable. To overcome the weaknesses of  $R^2$ , Henry Theil refined the  $R^2$  equation called adjusted  $R^2$ . The adjusted  $R^2$  shows that increasing the independent variables will further reduce the adjusted  $R^2$  value. The adjusted  $R^2$  value can still increase if the absolute value of the added variable is greater than 1, the greater the adjusted  $R^2$  value the better the model

## 4. RESULTS AND DISCUSSION

In this study, the we describes this research model with a diagram that will be used as a benchmark for the discussion to be written. The research model is basically an abstraction value of the variables being studied. In this case the research model is in accordance with the title of this study. Formulation of variables with their relevance as formulated in the diagram as follows:

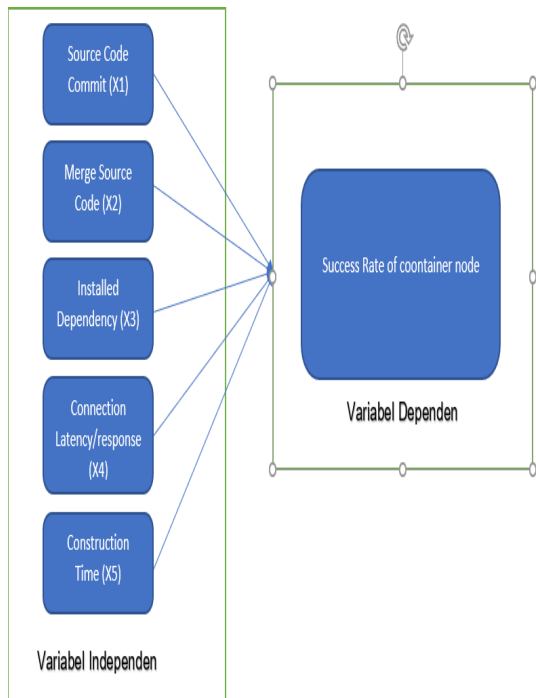


Figure 2. Research Methods

R. Descriptive Statistics Analysis

In order to test the descriptive statistical analysis, we will discuss the stages of classification of data based on the type of operating system used by each nodes to do automate Continuous Delivery / Continuous Integration process on existing container microservices. Data classification can be seen in the table:

No	Variable	Frekuensi	Presentase (%)
1	Jenis Sistem Operasi		
	prd-centos7-autorelease-8c-32g	1	1.2
	prd-centos7-builder-2c-1g	2	2.5
	prd-centos7-builder-2c-2g	5	6.3
	prd-centos7-builder-2c-8g	13	16.4
	prd-centos7-builder-4c-16g	2	2.5
	prd-centos7-builder-4c-4g	42	53.1
	prd-centos7-builder-8c-8g	5	6.3
	prd-centos7-docker-1c-4g	1	1.2
	prd-centos7-docker-2c-8g	1	1.2
	prd-centos7-robot-2c-8g	5	6.3
	prd-queue-intque-2c-1g	2	2.5

Table 2 : Data Frequentation based on OS

On table , it can be seen that from 79 microservices container nodes which are used as sampled in this study were divided based on the type of operating system used, the classification withinin the distribution based on operating systems is as follows, prd-centos7-autorelease-8c-32g as many as 1 node or 1.2%, prd-centos7-builder-2c-1g as much as 2 or 2.5%, prd-centos7-builder-2c-2g as much as 5 or 6.3%, prd-centos7-builder-2c-8g as many as 13 or 16.4 %, prd-centos7-builder-4c-16g as much as 2 or 2.5%, prd-centos7-builder-4c-4g as many as 42 or 53.1%, prd-centos7-builder-8c-8g as many as 5 or 6.3%, prd-centos7 -docker-1c-4g as many as 1 or 1.2%, prd-centos7-docker-2c-8g as many as 1 or 1.2%, prd-centos7-robot-2c-8g as much as 5 or 6.3%, prd-queue- intque-2c-1g as much as 2 or 2.5%.

The data is divided by straight benchmarks on JASP taken on data as many as 79 container nodes used in the Opendaylight project. Each microservices container is a part of the operating system on the existing node.

S. Hypothesis

In this study, we conducted a study of the analysis of the processes of use and implementation of container-based microservices architecture that existed in the development of the Opendaylight project, which is an open software based network defined software. In this criterion the data population will be taken to the success rate in applying container microservices by using data samples that have been published in Bitergia Analytics Opendaylight in the period from 1 October 2017 - 1 October 2018. The total population taken as samples is 79 microervices containers which has 5 influence factors in the successful implementation when automating code sources with CI / CD on the Jenkins master server. To get the required criteria, the hypothesis is set as follows :

- The number of microservices container nodes on each sub-project that is in the Opendaylight development statistics uses published data on tenants at 1 October 2017 - 1 October 2018. The number of successes in microservices container implementation in the last 1 year is written as primary data regardless of the number of developers and other Opendaylight development statistics. Broadly speaking, the data is only based on Jenkins node creation for each success process in the sub component component of Opendaylight.

- The number of nodes examined by the data has a composition and classification of operating systems for different container microservices between other nodes.
- The number of independent variables (X) for the Y variable has the obligation to determine the feasibility level of applying the microservices Opendaylight container system on the production scale.
- Independent variables together have a high degree of influence on the level of success of the implementation of container microservices architecture in Opendaylight.

T. Assumption Test Analysis

The results of the regression are interpreted, it is first tested whether there is a violation of the assumptions of multiple linear regression. In this study, testing of Double Correlation (R) and Determination Coefficient Test (R<sup>2</sup>), Simultaneous Significance Test (-F Statistical Test), Significance Test (-T Statistical Test), Assumption Test, Multicollinearity Test, Normality Test and Heteroscedasticity Test.

U. Normality Test

Normality test shows that the dependent variable and the independent variable are in the regression model. The normality test aims to test whether the sample used has a normal distribution or not. In a linear regression model, this assumption is indicated by the error value that is normally distributed. A good regression model is a regression model that has a normal or near normal distribution, so it is feasible to do statistical testing. Data normality testing uses the Test of Normality Kolmogorov-Smirnov in the JASP program. According to Singgih Santoso (2012: 293) the basis for decision making can be based on probability (Asymtotic Significance).

		Unstandardized Residual
N		76
Normal Parameters <sup>a,b</sup>	Mean	.0000000
	Std. Deviation	875.38671761
Most Extreme Differences	Absolute	.174
	Positive	.174
	Negative	-.131
Test Statistic		.174

Figure 3. Normality Test

V. Autocorrelation Test

Autocorrelation can be interpreted to show that there is a correlation between the error with the previous period error which in the classical assumption this should not happen. The autocorrelation test was carried out using the Durbin Watson method found in the JASP application program. Testing for the presence or absence of autocorrelation was performed using the Durbin-Watson method. The way to detect autocorrelation occurs in the regression analysis model using Durbin-Watson.

Decision criteria:

- If the value of Durbin Watson  $d < d_u$  or  $(4 - d_u) < d_u$ ,  $H_0$  is rejected, there is a positive or negative autocorrelation value.
- If the value of Durbin Watson  $d_u < d < 4 - d_u$ ,  $H_0$  is accepted, there is no positive or negative autocorrelation.

Model	R	R <sup>2</sup>	Adjusted R <sup>2</sup>	RMSE	Change F	Change df	df2	p	Durbin-Watson	
1	0.279	0.078	0.014	277.0	0.078	1.228	5	73	0.305	1.615

Figure 4. Autocorrelation Test

W. Multicollinearity Test

Multicollinearity test can aim to test whether the regression results found a correlation value between independent variables between one and the other variables. Multicollinearity can show that between the values of independent variables that have a very strong direct correlation (correlation). Tolerance is



the value of 1-R<sup>2</sup> from regression compared to that value with the rest of the other independent variables. Tolerance value approaching 0 indicates the existence of collinearity between an independent variable and the rest of the other independent variables. Another indicator of kolinieritas is Variance Inflation Factor (VIF) which is the reverse (reciprocal) of the tolerance value. The usual limit is 0.1 for tolerance, which means the limit of 10 for VIF (Hair et, 1998).

Coefficients							
Model	Unstandardized	Standard Error	Standardized	t	p	Collinearity Statistics	
						Tolerance	VIF
1 (Intercept)	190.673	69.315		2.751	0.007		
Source Code Commit	-0.379	0.255	-0.183	1.489	0.141	0.837	1.195
Merge Code (Avg)	-2.111	2.197	-0.165	0.961	0.340	0.431	2.322
Installed Dependency	0.128	0.137	0.111	0.933	0.354	0.895	1.118
Latency to Master Node	0.526	6.493	0.011	0.081	0.936	0.651	1.536

Figure 5. Multicollinearity Test

X. Heteroscedasticity Test

Heteroscedasticity is where the situation is not constant variance. To detect the presence or absence of heteroscedasticity testing using the Glejser method, a comparison between the sig-t value and 0.05 was carried out. If Sig-t a calculated value smaller than 0.05 then heteroscedasticity will occur, and vice versa. If sig-t\_count is greater than 0.05 then heteroscedasticity will not occur. The Glejser test results can be shown in the following table:

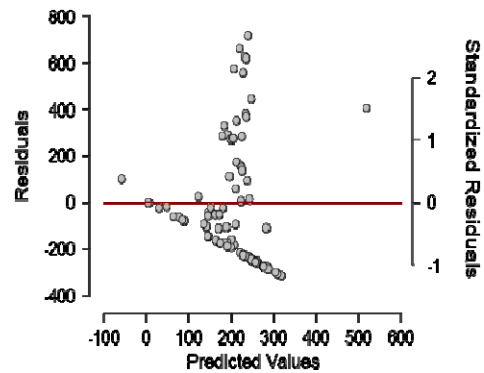


Figure 6. Heteroscedasticity Test

Based on the results of the scatter plot above it is known that data retrieval does not show a certain pattern. The scattering of data spread randomly so that we concluded that there was no residual heteroscedasticity problem. Similarly, the Glejser test results show that the significance values for each independent variable in the regression model equation for the absolute residual value are > 0.05. Thus it can be concluded that the data is homokedastic.

Y. Significance Hypothesis Test (Test F)

The F-test is used to test whether all of the independent variables together influence significantly on the dependent variable.

H0 :  $\beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0$

Source code commit, merge source code, installed dependency, connection latency and construction time together do not affect the level of implementation of the container microservices architecture with CI / CD automation on Jenkins nodes in the Opendaylight project development environment..

Ha :  $\beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0$

Source code commit, merge source code, installed dependency, connection latency, and construction time together make the influence the level of implementation of the container microservices architecture with CI / CD automation on Jenkins

nodes in the Opendaylight project development environment.

The results of the Significance Test using JASP can be seen in the following figure :

ANOVA						
Model		Sum of Squares	df	Mean Square	F	p
1	Regression	471274	5	94255	2.428	0.025
	Residual	5.603e+6	73	76748		
	Total	6.074e+6	78			

Figure 7. Significance Test

Based on table 4.3, the F-count value is 5.905 with a Sig value of 0.000. This indicates that the calculated value is greater than Ftable 2.471 and the Sig value is smaller than 0.05. Thus H0 is rejected and Ha is accepted. As a result, the source code commit, merge source code, installed connection latency and construction time dependencies have had an effect on the success of container implementation for DevOps in the development of the Opendaylight project carried out in October 2017 until October 2018. Referring to this F test the implementation of microservices architecture should be implemented by adhering to the criteria contained in the use of containers in Docker or the application of multiple container orchestration with quarantines. So that it can reduce the resources that exist in the production environment.

Z. Test - T

The t-test is used to determine the effect of partial independent variables on the dependent variable. This test is by comparing the probability value or p-value (sig-t) with a significance level of 0.05. If the p-value is smaller than 0.05 then Ha is accepted, and vice versa if the p-value is greater than 0.05 then Ha is rejected.

Coefficients						
Model		Unstandardized	Standard Error	Standardized	t	p
1	(Intercept)	190.673	69.313		2.751	0.007
	Source Code Commit	-0.379	0.255	-0.183	1.489	0.041
	Merge Code (Avg)	-2.111	2.197	-0.165	-0.961	0.340
	Installed Dependency	0.128	0.137	0.111	0.933	0.354
	Latency to Master Node	0.526	6.493	0.011	0.081	0.936
	Construction Time	7.304	4.249	0.299	1.719	0.090

Figure 8. T-test result

The conclusions of the results of the T Test above are as follows::

- The influence of the source code commit towards success factors  
Based on the table can be obtained T-count value of 1.48 with Sig value of 0.141. This shows that the t-count value is smaller than the value of t-table 2.37 and the sig value is smaller than 0.05. Thus H0 is rejected and Ha is rejected.
- The influence of merge source code towards success factors  
Based on table 4.9 can be obtained by the results of the t-count value of -0.961 and the Sig value of 0.340. This shows that the T-count value is smaller than the t-table value 2.37 and the Sig 0.340 value is greater than 0.05. Thus it can be concluded that H0 is rejected and Ha is accepted.
- The influence of installed dependency towards success factors  
Based on table 4.9 can be obtained by the results of the t-count value of 0.933 and the Sig value of 0.354. This shows that the T-count value is smaller than the t-table value 2.37 and the Sig 0.354 value is greater than 0.05. Thus it can be concluded that H0 is rejected and Ha is accepted..
- The influence of connection latency or response time towards success factors  
Based on table 4.9 can be obtained by the results of the t-count value of 0.081 and the Sig value of 0.936. This shows that the T-count value is smaller than the value of t-table 2.37 and the Sig 0.936 value is greater than 0.05. Thus it can be concluded that H0 is rejected and Ha is accepted..
- The influence of construction time towards success factors  
Based on table 4.9 can be obtained with the results of the t-count value of 1.719 and the Sig value of 0.936. This shows that the T-count value is smaller than the t-table value 2.37 and the Sig 0.090 value is greater than 0.05. Thus it can be concluded that H0 is rejected and Ha is accepted.

Result

Through the results of the t-test obtained, it can be concluded that each value of the independent variables has each of the different value roles as outlined, among others; The source code commit factor has no significant effect on the implementation of microservices architecture,

because it can be concluded that the number of commit values for the code that occurs on the git server only affects every local repository that exists for the development process, but does not affect the master the node on the Jenkins server when the image container encapsulation process takes place; merge factor source code has an influence than the success factor of microscope architecture implementation, this can be interpreted by the number of each repository contained in the git sever when encapsulated into a container image package for microservices system then the number of branches that determine the jenkins master node can do create image or not.

## 5. CONCLUSION

As the results of the test and previous analysis it was determined that, the source code commit variable did not have a significant effect on the success ratio of microservices container implementation while the other variable can be determined as the major factors of sucess towards the implementation of microservices container-based architecture. Test the regression model shows that the level of the ratio is classified as invalid and low. This can be interpreted that regardless of the source code that is automated on the Git server it does not affect the implementation of the container that was successfully deployed. Therefore if the production environment is used, the sysadmins or DevOps engineers can automate the source code commit on an ongoing basis..

## REFERENCES:

- [1] Nuha Alshuqayran, Nour Ali and Roger Evans. A Systematic Mapping Study in Microservice Architecture. In IEEE 9th International Conference on Service-Oriented Computing and Applications (2016).
- [2] Muhammad Waseem, Peng Liang. Microservices Architecture in DevOps. In 24th Asia-Pacific Software Engineering Conference Workshops (2017).
- [3] Kratzke, N. (2015). About Microservices, Containers and their Underestimated Impact on Network Performance. In Cloud Computing Conf (2015).
- [4] Oleg Mironov. DevOps Pipeline with Docker. Helsinki Metropolia University of Applied Sciences, Thesis (2018).
- [5] Dharmendra Shadija and Mo Rezai, Richard Hill. Towards an Understanding of Microservices. In Proceedings of the 23rd International Conference on Automation & Computing, University of Huddersfield, Huddersfield, UK, 7-8 September (2017).
- [6] Zhongxiang Xiao, Inji Wijegunaratne, Infosys Australia, Xinjian Qiang. Reflections on SOA and Microservices. In 4th International Conference on Enterprise Systems (2016).