Jurnal Teknologi Informasi dan Pendidikan
Volume 17, No. 2, September 2024
https://doi.org/10.24036/jtip.v17i2.869

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

Comparison of Automation Deployment Implementation on Google Cloud Virtual Machine Using Deployment Manager and Terraform

Naila Ardelia[1], Lindawati[1*], Nurhajar Anugraha[1]
Electrical Engineering, Politeknik Negeri Sriwijaya, Palembang, Indonesia
*Corresponding Author: lindawati@polsri.ac.id

Article Information

Article history:

No. 869
Rec. July 16, 2024
Rev. January 13, 2025
Acc. January 14, 2025
Pub. January 15, 2025
Page. 529 – 542

Keywords:

- *Cloud Computing*
- *Automation Deployment*
- *Google Cloud Platform*
- *Terraform*

ABSTRACT

Software development has significantly transformed in recent years, with organizations increasingly turning to cloud computing infrastructure to speed up and ease the development and implementation processes. Up until now, there are few studies testing the efficiency and performance of deployment automation using Google Cloud Deployment Manager, which is essentially a built-in infrastructure deployment service provided by Google Cloud Platform. Most studies tend to prefer using open-source software such as Terraform, Ansible, and Kubernetes. This research aims to compare the implementation of deployment automation using Google Cloud Deployment Manager and Terraform. Three parameters are used to compare the results of automation deployment implementation: deployment efficiency, website performance, and cost efficiency. The test results indicate that Google Cloud Deployment Manager outperforms Terraform in all three test parameters. Specifically, Google Cloud Deployment Manager demonstrated superior deployment efficiency, enhanced website performance, and better cost efficiency. Thus, it is concluded that Google Cloud Deployment Manager is a more effective solution for deployment automation compared to Terraform.

How to Cite:

Ardelia, N., et al. (2024). Comparison of Automation Deployment Implementation on Google Cloud Virtual Machine Using Deployment Manager and Terraform. Jurnal Teknologi Informasi Dan Pendidikan, 17(2), 529-542. https://doi.org/10.24036/jtip.v17i2.869

1. INTRODUCTION

The development of software has undergone significant transformation in recent years, with more organizations shifting to cloud computing infrastructure to accelerate and simplify the development and implementation processes. Cloud computing is a method that provides shared computing resources, including applications, computing, storage, networking, development, and deployment platforms, as well as business processes [1]. It is often described as the abstraction of web-based computing resources and services, enabling system developers to implement complex systems through web-based platforms [2]. The concept of cloud computing stems from the idea that data processing and storage can be performed more efficiently in large-scale computing and storage systems accessible via the internet. Cloud computing is a model that facilitates on-demand computing access, with sufficient network access to quickly share pooled resources with minimal management effort [3].

There are four fundamental characteristics of cloud computing that unify all platforms into a single integrated platform: on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service[4]. On-demand self-service allows users to independently define computing capabilities as needed, broad network access means that cloud computing can be accessed anywhere and anytime, resource pooling is the practice of aggregating computing resources and making them available as a single entity, rapid elasticity is the ability of cloud infrastructure to adjust computing resources quickly and automatically according to user needs, and measured service allows users to pay only for the resources they use.

When building cloud computing infrastructure, users must go through several stages such as filling out forms, selecting the operating system, and configuring other services. Managing and developing multiple servers manually requires considerable time and effort, making automation necessary. Automating the deployment of complex distributed applications is essential today, as it allows for the full utilization of cloud computing's potential. Manually deploying complex systems is prone to errors, time-consuming, and expensive[5].

There are numerous technologies and platforms available for automation deployment. In study [6], automation deployment is conducted using Novel Google Kubernetes and Microsoft Kubernetes to compare the performance of these two tools. Study [7] compares the implementation of automation deployment using Docker Swarm and Kubernetes. Study [8] investigates the comparison of implementing Terraform and Cloudify. Studies [5] and [9] examine the implementation of Terraform.

There is a wide range of cloud platforms available, such as Google Cloud, Amazon Web Services, Microsoft Azure, and others. In study [10], automation deployment is performed using HA Proxy and Ansible on the AWS platform. Automation deployment can also be conducted on the Microsoft Azure platform, as applied in study [11]. Besides comparing deployment tools, there are also studies that compare the implementation of AWS Lambda on the AWS platform and Google Cloud Function on the Google Cloud Platform, as seen in study [12].

Previous studies have demonstrated the importance of automation deployment in managing and developing cloud resources more efficiently in terms of time, effort, and cost. The frequently used platforms are Google Cloud and AWS. However, unlike the documentation of AWS services, which is not publicly disseminated, the documentation for Google Cloud services is much more comprehensive and publicly available. From previous research, most automation deployments have been conducted using Terraform. There are still few studies discussing automation deployment using Google Cloud Deployment Manager, which is the built-in tool provided by the Google Cloud Platform.

Therefore, this research aims to compare the implementation of the two software services: Terraform and Google Cloud Deployment Manager, by evaluating them based on configuration and deployment efficiency, the performance of the target website, and the costs involved in managing resources. Hence, this research is titled "Comparison of Automation Deployment Implementation on Google Cloud VM Using Deployment Manager and Terraform."

2. RESEARCH METHOD

Research methodology based on literature review by collecting reference materials from books, journals, articles, google cloud documentations, and other internet sources.

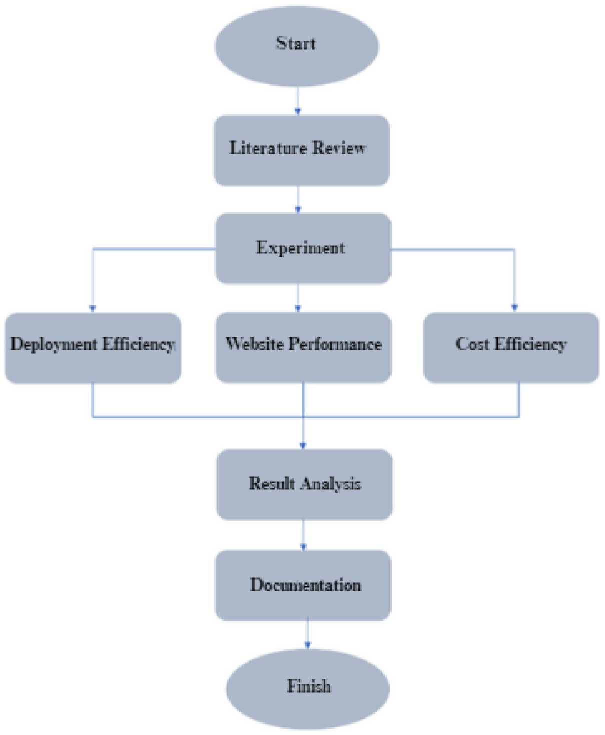


Figure 1. Research Flow

The research flow is explained as follows:

1. Start
2. Literature Review: systematically gathers and synthesizes existing research to provide a comprehensive understanding of the topic[13].
3. Experiment: This stage tests the deployment with the test subject being an inventory stock website. The testing is conducted based on three parameters, namely deployment process efficiency, website performance, and cost efficiency.
4. Result Analysis: After the testing phase, the results are analyzed to derive conclusions from the comparative implementation.
5. Documentation: Documenting the findings.
6. Finish.

2.1. Software Used

Research methodology based on literature review by collecting reference materials from books, journals, articles, google cloud documentations, and other internet sources. The software and applications utilized in this research are listed in Table 1.

Table 1. Software List

No.	Software	Function
1.	Visual Studio Code Version 1.87.2	Building and editing program
2.	Laravel Version 8	A framework that helps maximize the use of the PHP programming language, emphasizes flexibility and simplicity in its design[14].
3.	PHP Version 8.3.4	Used as a server-side script in web development and is embedded within HTML documents[15].
4.	MySQL	Organizes and manages database queries, enabling websites to become more dynamic and multifunctional[16]
5.	Apache	generates accurate web pages for users based on the PHP code written by web developers or users [16]
6.	Google Cloud Platform	Cloud computing platform that offers on-demand internet-based support[17]
7.	Google Cloud Deployment Manager	It is GCP's built-in standard deployment automation tool used in this research.
8.	Terraform	An open-source software which gives an infrastructure as code (IaC) interface to cloud providers [18] that used for automation deployment tool.

2.2. Test Object

The test object in this research is an inventory stock website, as shown in Figure 2.

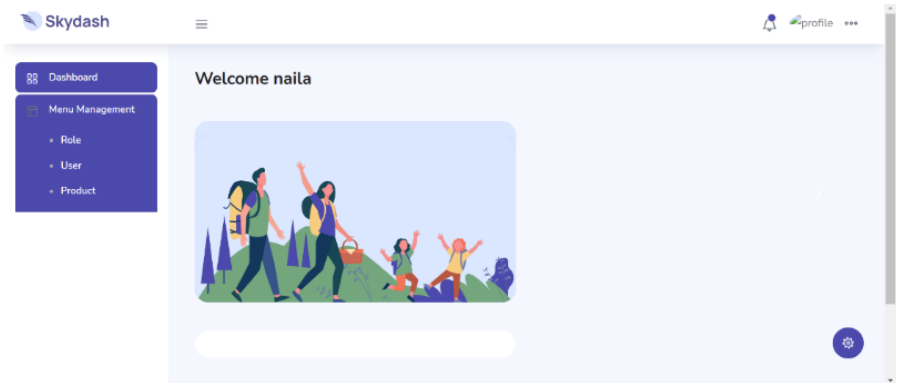


Figure 2. User Interface

2.3. Observed Parameters

The testing process involves three parameters: deployment efficiency, website performance, and cost efficiency.

1) Deployment Process Efficiency
   For testing the efficiency of the deployment process, two benchmarks are used: ease of command deployment and the duration of the deployment process[6].
2) Website Performance
   Website performance testing utilizes two tools: GTmetrix, an open-source tool, and PageSpeed Insights, a Google tool, to measure the performance level of the website being tested. Also, PageSpeed Insight uses the same rules that GTmetrix uses in providing value of a website[19].
3) Cost Efficiency
   Cost efficiency testing compares the service costs of the two automation deployment methods using cloud billing reports that show Google Cloud service charges[20].

3. RESULTS AND DISCUSSION

3.1. GCDM Deployment Process

The deployment process begins by preparing the infrastructure on Google Cloud Platform using Compute Engine. The first deployment is carried out with Google Cloud Deployment Manager (GCDM); if it fails, the process is repeated. If successful, deployment is continued using Terraform in a similar manner. After both deployments are successful, deployment speed, website performance, and service costs are tested to determine which automation deployment method is more effective and efficient between GCDM and Terraform.

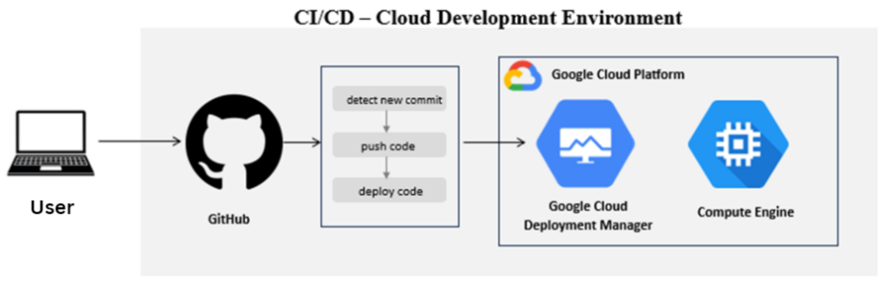


Figure 3. GCDM Deployment Flow

This system uses CI/CD methodology that facilitates rapid software updates while ensuring system stability and security[21]. First, the administrator builds the website code and pushes it to the GitHub repository. Next, the admin prepares the infrastructure on Google Cloud Platform with Compute Engine services as shown in Figure 3. Deployment is done using Google Cloud Deployment Manager by pushing the code from GitHub again.

3.2. Terraform Deployment Process

Automation deployment uses CI/CD methodology that facilitates rapid software updates while ensuring system stability and security[21]. The admin builds the website code and pushes it to GitHub. The infrastructure is prepared on Google Cloud Platform using Compute Engine. Before deployment on Compute Engine, Terraform pushes the code from GitHub to Compute Engine, so the website code is not directly deployed on Google Cloud Platform but through Terraform first.

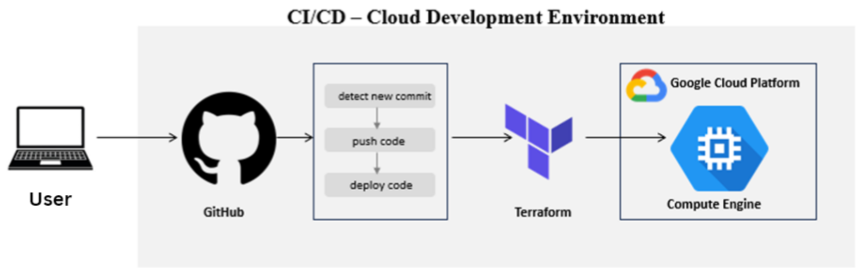


Figure 4. Terraform Deployment Flow

3.3. Testing Deployment Efficiency Parameters

Before the deployment process using Google Cloud Deployment Manager, the initial step required is to install the Google Cloud SDK (Software Development Kit) first. Then, the deployment command can be executed on the Google Cloud SDK Shell.

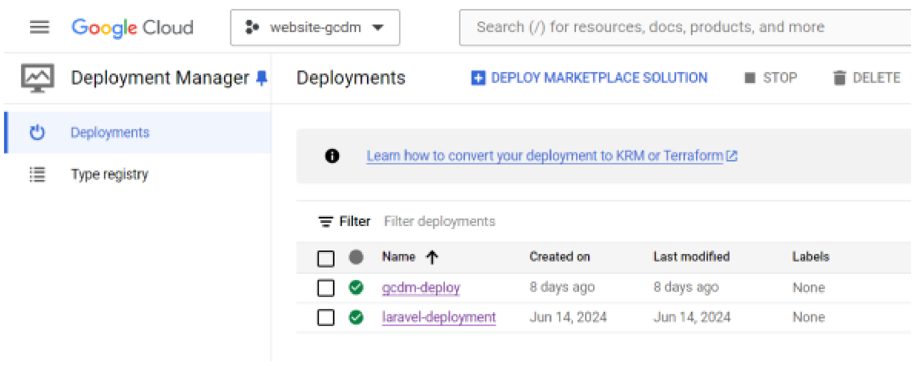
Table 2. Test Result

Trial	Duration (s)
1	21
2	21
3	18
4	19
5	20
6	21
7	18
8	22
9	19
10	20

Table 2 shows ten deployment trials on GCDM to obtain the average deployment duration. The deployment durations listed in the table are presented in seconds. The average deployment time calculation is as follows:

$$\frac{21 + 21 + 18 + 19 + 20 + 21 + 18 + 22 + 19 + 20}{10} = 19{,}9\ s$$

The Terraform deployment output can be seen on the Deployment Manager as shown in Figure 5.


Google Cloud
website-gcdm
Search (/) for resources, docs, products, and more
Deployment Manager
Deployments
DEPLOY MARKETPLACE SOLUTION
STOP
DELETE
Deployments
Type registry
Learn how to convert your deployment to KRM or Terraform
Filter
Filter deployments

Name	Created on	Last modified	Labels
gcdm-deploy	8 days ago	8 days ago	None
laravel-deployment	Jun 14, 2024	Jun 14, 2024	None

Figure 5. GCDM Deployment Output

For deployment using Terraform, it is necessary to install Terraform first. However, deployment can be done on the Visual Studio Code terminal alone, so users do not need to open a special command prompt for the deployment process.

Table 3. Test Result

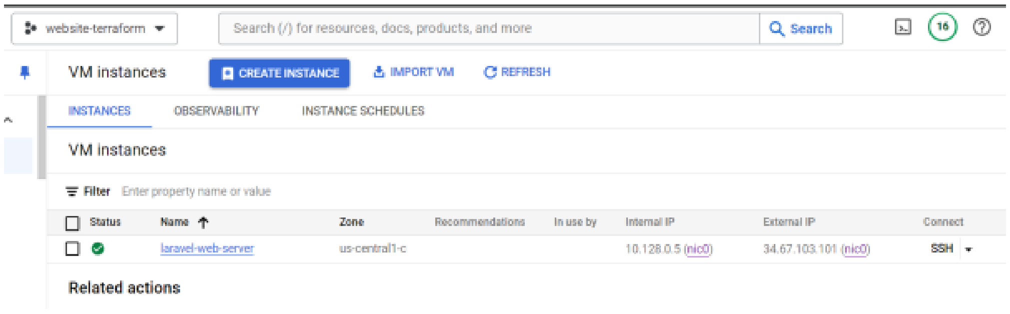
Trial	Duration (s)
1	36
2	34
3	34
4	35
5	33
6	34

7	31
8	35
9	32
10	31

Table 3 shows ten deployment trials on Terraform to obtain the average deployment duration. The deployment durations listed in the table are presented in seconds. The average deployment time calculation is as follows:

$$\frac{36+34+34+35+33+34+31+35+32+31}{10} = 33{,}5\ s$$

The Terraform deployment output can be seen on the Compute Engine Instance as shown in Figure 6.



Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
✓	laravel-web-server	us-central1-c			10.128.0.5 (nic0)	34.67.103.101 (nic0)	SSH

Figure 6. Terraform Deployment Output

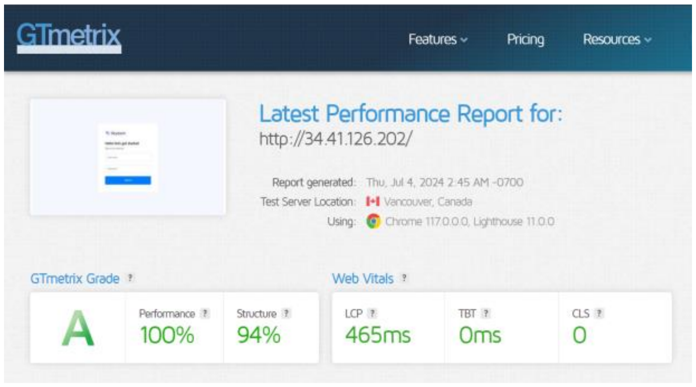
3.4. Testing Website Performance Parameters

In the website performance testing, website feasibility categories are applied based on Table 4.

Table 4. Website Feasibility

Percentage	Description
0%-20%	Very Poor
21%-40%	Poor
41% - 60%	Average
61%-80%	Good
81-100%	Excellent

Performance comparison using GTmetrix includes parameters such as performance, structure, and LCP (Largest Contentful Paint). Performance measures how well a website runs based on user experience. Structure measures how well a website is built for optimal performance. LCP measures how quickly substantial content on the website can be consumed by users.



Latest Performance Report for:
http://34.41.126.202/

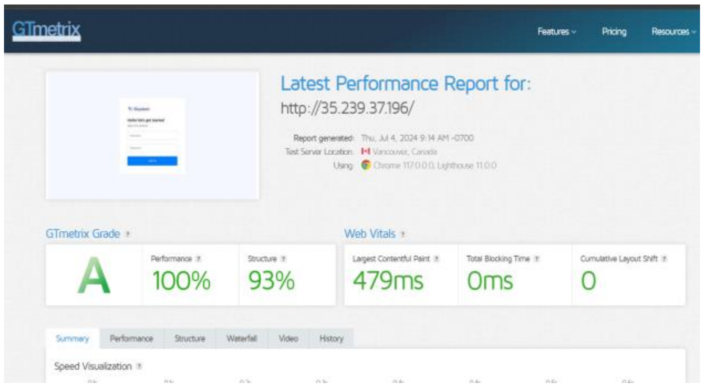
Report generated: Thu, Jul 4, 2024 2:45 AM -0700
Test Server Location: Vancouver, Canada
Using: Chrome 117.0.0.0, Lighthouse 11.0.0

GTmetrix Grade: A
Performance: 100%
Structure: 94%

Web Vitals
LCP: 465ms
TBT: 0ms
CLS: 0

Figure 7. Test Result of GCDM Website Performance

Figure 7 shows the website performance test results using the GTmetrix tool. The test results indicate a performance score of 100%, a structure score of 94%, and an LCP of 465 ms. Based on the feasibility level in Table 4, the GCDM website has excellent performance with a performance score of 100%.



Latest Performance Report for:
http://35.239.37.196/

Report generated: Thu, Jul 4, 2024 9:14 AM -0700
Test Server Location: Vancouver, Canada
Using: Chrome 117.0.0.0, Lighthouse 11.0.0

GTmetrix Grade: A
Performance: 100%
Structure: 93%

Web Vitals
Largest Contentful Paint: 479ms
Total Blocking Time: 0ms
Cumulative Layout Shift: 0

Figure 8. Test Result of Terraform Website Performance

Figure 8 shows the website performance test results using the GTmetrix tool. The test results indicate a performance score of 100%, a structure score of 93%, and an LCP of 479 ms. Based on the feasibility level in Table 4, the Terraform website has excellent performance with a performance score of 100%.

Table 5. Comparison Using GTmetrix

Parameters	*GCDM*	Terraform	Description
Performance	100%	100%	Equal
Structure	94%	93%	GCDM is 1% better than Terraform
LCP	465 ms	479 ms	GCDM is 14 ms faster than Terraform

From table 5, it shows Google Cloud Deployment Manager has better performance than the test results on Terraform.

For performance testing using PageSpeed Insights, several parameters are applied, including Performance, Accessibility, Best Practice, and SEO (Search Engine Optimization). Performance measures website capability based on several metrics such as First Contentful Paint, Largest Contentful Paint, and others. Accessibility measures the ease of access to a website for a wide audience. Best Practice measures the website's ability to handle traffic quickly and responsively. SEO measures whether the website adheres to search engine optimization standards.

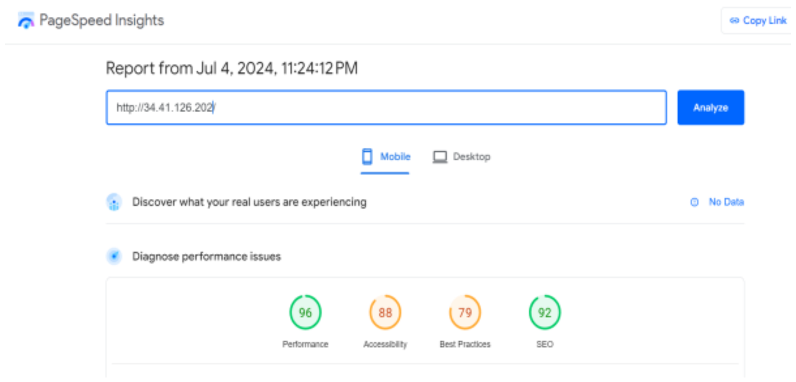


Figure 9. Test Result of GCDM Website Performance

Figure 9 shows the website performance test results for GCDM using the PageSpeed Insights tool. The test results indicate a performance score of 96%, accessibility score of 88%, best practices score of 79%, and SEO score of 92%. Based on the feasibility level in Table 4, the GCDM website has excellent performance with a performance score of 96%.

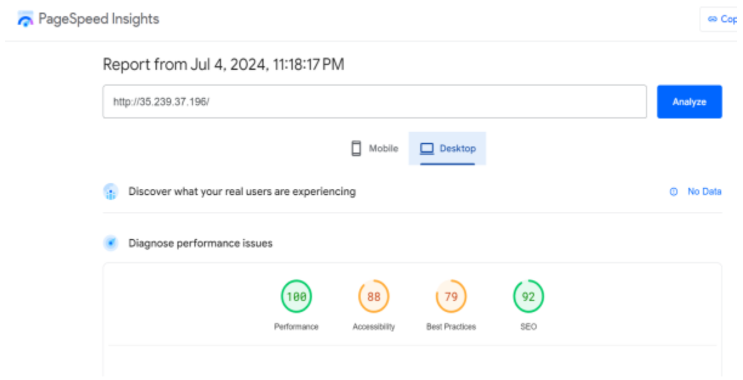


Figure 10. Test Result of Terraform Website Performance

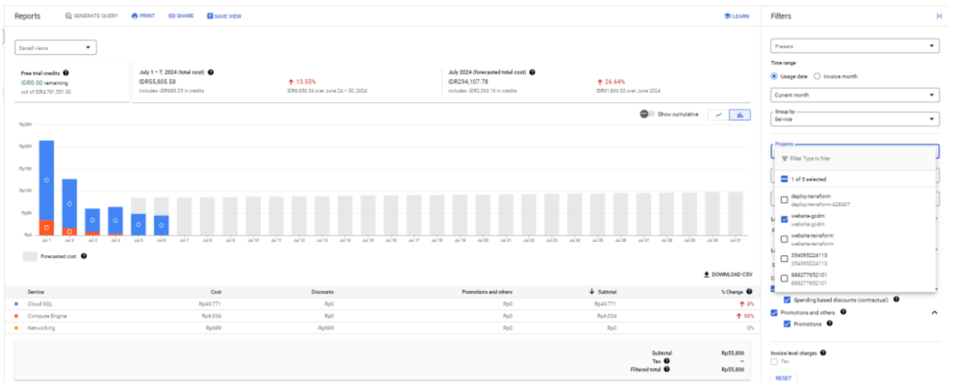
Figure 10 shows the website performance test results for Terraform using the PageSpeed Insights tool. The test results indicate a performance score of 100%, accessibility score of 88%, best practices score of 79%, and SEO score of 92%. Based on the feasibility level in Table 4, the Terraform website has excellent performance with a performance score of 100%.

Table 6. Comparison Using PageSpeed Insights

Parameter	GCDM	Terraform	Description
Performance	96%	100%	Terraform is 4% better that GCDM
Accessibility	88%	88%	Equal
Best Practice	79%	79%	Equal
SEO	92%	92%	Equal

Table 6 presents the performance comparison using PageSpeed Insights with parameters including performance, accessibility, best practice, and SEO. Based on these parameters, Terraform shows better performance than the test results on Google Cloud Deployment Manager.

3.5. Testing Website Performance Parameters



Free trial credits
IDR0.00 remaining
out of IDR4,781,551.00

July 1 – 7, 2024 (total cost)
IDR55,805.58
Includes -IDR689.35 in credits
↑ 13.55%
IDR6,658.54 over June 24 – 30, 2024

July 2024 (forecasted total cost)
IDR294,107.78
Includes -IDR2,363.16 in credits
↑ 26.64%
IDR61,866.52 over June 2024

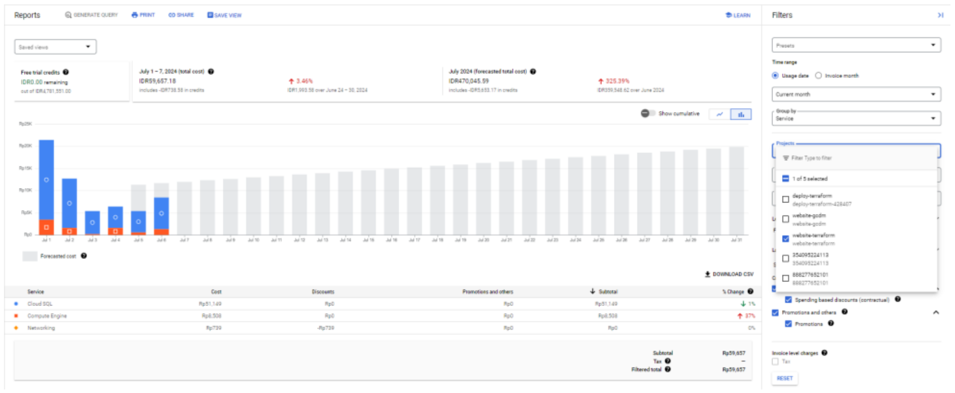
Service	Cost	Discounts	Promotions and others	Subtotal	% Change
Cloud SQL	Rp49,771	Rp0	Rp0	Rp49,771	↑ 8%
Compute Engine	Rp6,034	Rp0	Rp0	Rp6,034	↑ 93%
Networking	Rp689	-Rp689	Rp0	Rp0	0%

Subtotal Rp55,806
Tax –
Filtered total Rp55,806

Time range: [✓ Usage date] Invoice month
Current month
Group by: Service
Projects: 1 of 5 selected
☐ deploy-terraform (deploy-terraform-428407)
☑ website-gcdm (website-gcdm)
☐ website-terraform (website-terraform)
☐ 354095224113 (354095224113)
☐ 888277652101 (888277652101)
☑ Spending based discounts (contractual)
☑ Promotions and others
☑ Promotions
Invoice level charges
☐ Tax

Figure 11. GCDM Billing

Figure 11 shows the cost usage details for the website project using Google Cloud Deployment Manager, amounting to Rp 55,806 as of July 7, 2024, and counted from July 1, 2024. This cost includes IDR 49,771 for SQL services, IDR 6,034 for VM Instance Compute Engine, and IDR 689 for network costs.



Reports

Free trial credits: IDR0.00 remaining, out of IDR4,781,551.00

July 1 – 7, 2024 (total cost): IDR59,657.18, includes -IDR738.58 in credits; ↑ 3.46%, IDR1,993.58 over June 24 – 30, 2024

July 2024 (forecasted total cost): IDR470,045.59, includes -IDR5,653.17 in credits; ↑ 325.39%, IDR359,548.62 over June 2024

Service	Cost	Discounts	Promotions and others	Subtotal	% Change
Cloud SQL	Rp51,149	Rp0	Rp0	Rp51,149	↓ 1%
Compute Engine	Rp8,508	Rp0	Rp0	Rp8,508	↑ 37%
Networking	Rp739	-Rp739	Rp0	Rp0	0%

Subtotal Rp59,657
Tax –
Filtered total Rp59,657

Figure 12. Terraform Billing

Figure 12 shows the cost usage for the website project using Terraform, amounting to IDR 59,657 as of July 7, 2024, and counted from July 1, 2024. This cost includes IDR 51,149 for SQL services, IDR 8,508 for VM Instance Compute Engine, and IDR 739 for network costs.

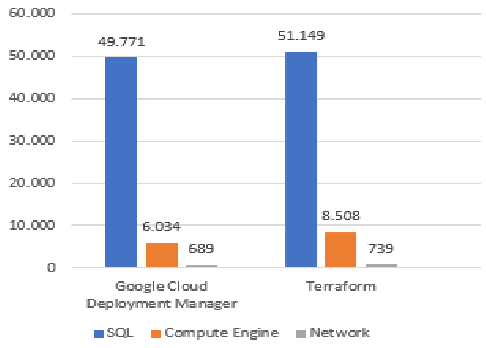


Figure 13. Billing Comparison

Based on the service cost comparison in Figure 13, it can be concluded that Google Cloud Deployment Manager is more cost-efficient.

4. CONCLUSION

From the research conducted, the following conclusions can be drawn regarding the comparison of automation deployment implementation on Google Cloud VM using Google Cloud Deployment Manager and Terraform:

1) Deployment Success: Deployment using both Google Cloud Deployment Manager and Terraform was successfully executed, as evidenced by the testing of each parameter.
2) Deployment Efficiency: In the first parameter test, which is deployment efficiency, Google Cloud Deployment Manager excels in terms of shorter average deployment

duration. However, in terms of ease of the deployment process, Terraform is superior as it does not require a special command prompt and can be run in the Visual Studio Code terminal alone.

3) Website Performance: In the second parameter test, which is website performance, Google Cloud Deployment Manager performs better by 1% when tested using GTmetrix. On the other hand, Terraform performs 4% better when tested using PageSpeed Insights.
4) Service Costs: In the third parameter test, which is service costs, Terraform incurs higher costs compared to Google Cloud Deployment Manager. Therefore, it can be concluded that in terms of service costs, Google Cloud Deployment Manager is more advantageous.
5) Overall Results: Based on the tests conducted using the parameters of deployment efficiency, website performance, and service costs, the results show that Google Cloud Deployment Manager performs better than Terraform.

ACKNOWLEDGEMENTS

The author would like to express sincere gratitude to all those who have provided support and assistance during this research, especially to my beloved parents, my brother, my academic advisor, and my friends. Thank you for all the guidance and support, which have been invaluable to the success of this research.

REFERENCES

[1] J. S. Hurwitz and D. Kirsch, *Cloud Computing for Dummies*. John Wiley & Sons., 2020.
[2] K. Jamsa, *Cloud Computing*. Jones & Bartlett Learning, 2022.
[3] D. C. Marinescu, *Cloud computing: theory and practice*. 2022.
[4] K. Latha and T. Sheela, "Block based data security and data distribution on multi cloud environment," *J Ambient Intell Humaniz Comput*, Jul. 2019, doi: 10.1007/s12652-019-01395-y.
[5] M. Wurster *et al.*, "Automating the Deployment of Distributed Applications by Combining Multiple Deployment Technologies," in *Proceedings of the 11th International Conference on Cloud Computing and Services Science*, SCITEPRESS - Science and Technology Publications, 2021, pp. 178–189. doi: 10.5220/0010404301780189.
[6] S. V. Kumar and M. K, "Estimating the Deployment Time for Cloud Applications using Novel Google Kubernetes Cloud Service over Microsoft Kubernetes Cloud Service," *J Pharm Negat Results*, vol. 13, no. SO4, 2022, doi: 10.47750/pnr.2022.13.S04.186.
[7] J. Shah and D. Dubaria, "Building Modern Clouds: Using Docker, Kubernetes &amp; Google Cloud Platform," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, 2019, pp. 0184–0189. doi: 10.1109/CCWC.2019.8666479.
[8] L. R. de Carvalho and A. Patricia Favacho de Araujo, "Performance Comparison of Terraform and Cloudify as Multicloud Orchestrators," in *2020 20th IEEE/ACM International Symposium on*

Cluster, Cloud and Internet Computing (CCGRID), IEEE, May 2020, pp. 380–389. doi: 10.1109/CCGrid49817.2020.00-55.

[9] Y. Mansouri, V. Prokhorenko, and M. A. Babar, "An automated implementation of hybrid cloud for performance evaluation of distributed databases," *Journal of Network and Computer Applications*, vol. 167, p. 102740, Oct. 2020, doi: 10.1016/j.jnca.2020.102740.

[10] S. Sokolov, O. Idiriz, M. Vukadinoff, and S. Vlaev, "Scaling and Automation in Cloud Deployments of Enterprise Applications," *JOURNAL OF Engineering Science and Technology Review*, 2020.

[11] Y. Li, Z. Han, Q. Zhang, Z. Li, and H. Tan, "Automating Cloud Deployment for Deep Learning Inference of Real-time Online Services," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, IEEE, Jul. 2020, pp. 1668–1677. doi: 10.1109/INFOCOM41043.2020.9155267.

[12] M. Malawski, A. Gajek, A. Zima, B. Balis, and K. Figiela, "Serverless execution of scientific workflows: Experiments with HyperFlow, AWS Lambda and Google Cloud Functions," *Future Generation Computer Systems*, vol. 110, pp. 502–514, Sep. 2020, doi: 10.1016/j.future.2017.10.029.

[13] H. Snyder, "Literature review as a research methodology: An overview and guidelines," *J Bus Res*, vol. 104, pp. 333–339, Nov. 2019, doi: 10.1016/j.jbusres.2019.07.039.

[14] D. Novaliendry and V. D. Puteri, "E-Retail Percetakan Anambaleh Desain Menggunakan Framework Laravel," *Jurnal Teknologi Informasi dan Pendidikan*, vol. 13, no. 1, 2020.

[15] T. Sriwahyuni, O. Oktoria, and I. P. Dewi, "PENGEMBANGAN SISTEM INFORMASI MANAJEMEN PARIWISATA BERBASIS WEB," *Jurnal Teknologi Informasi dan Pendidikan*, vol. 12, no. 1, pp. 92–99, Mar. 2019, doi: 10.24036/tip.v12i1.184.

[16] T. Sriwahyuni and T. Surianto, "Design of Website-Based Information System for Psychology Service Unit of Universitas Negeri Padang Using YII2 Framework," *Jurnal Teknologi Informasi dan Pendidikan*, 2023.

[17] R. U. A. Sajid, S. Islam, A. B. K. Rakib, and A. Kaur, "Interpretation on the Google Cloud Platform and Its Wide Cloud Services," *International Journal of Security and Privacy in Pervasive Computing*, vol. 14, no. 1, pp. 1–7, Nov. 2022, doi: 10.4018/IJSPPC.313586.

[18] M. Zadka, *DevOps in Python*. Berkeley, CA: Apress, 2022. doi: 10.1007/978-1-4842-7996-0.

[19] D. Muriyatmoko and Aziz Musthafa, "Website Performance Testing Using Speed Testing Model: A Case of Reputable Indonesian Journals," *Teknik: Jurnal Ilmu Teknik dan Informatika*, vol. 2, no. 1, pp. 40–45, May 2022, doi: 10.51903/teknik.v2i1.120.

[20] O. Alzakholi, L. Haji, H. Shukur, R. Zebari, S. Abas, and M. Sadeeq, "Comparison Among Cloud Technologies and Cloud Performance," *Journal of Applied Science and Technology Trends*, vol. 1, no. 1, pp. 40–47, Apr. 2020, doi: 10.38094/jastt1219.

[21] Y. SKA and J. P, "A Study And Analysis Of Continuous Delivery, Continuous Integration In Software Development Environment," *J Emerg Technol Innov Res*, Sep. 2019.