CLOUD COMPUTING TECHNOLOGY PRACTICUM REPORT

IAAS (INSFRASTRUCTURE AS A SERVICE)
CLOUD COMPUTING TECHNOLOGY PRACTICUM
PLUG - H

By:

Nama : Sabila Khairina Saktiwati
NIM : 123220085

INFORMATICS STUDY PROGRAM
DEPARTMENT OF INFORMATICS
FACULTY OF INDUSTRIAL ENGINEERING
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"
YOGYAKARTA

2025

APPROVAL PAGE

PRACTICUM REPORT

CLOUD SQL AND CLOUD STORAGE
CLOUD COMPUTING TECHNOLOGY PRACTICUM
PLUG - H

By:

Sabila Khairina Saktiwati 123220085

Checked and approved by the Practicum Assistant of Cloud Computing Technology.
At the date of: ……

.

Approved by.

Practicum Assistant	**Practicum Assistant**
Muhammad Rafli	**Sayang Sani**
NIM 123210078	**NIM 123210044**

PREFACE

Praise and gratitude are expressed to the presence of God Almighty for His blessings and grace, which have enabled the completion of this practicum report. This report is prepared as part of fulfilling academic requirements and as a form of accountability for the practicum activities that have been carried out.

The author would like to express sincere appreciation to Muhammad Rafli, and Sayang Sani for their guidance and assistance throughout the practicum.

The author is fully aware that this report is still far from perfect. Therefore, constructive criticism and suggestions are highly expected for future improvements. Hopefully, this report can provide benefits to all relevant parties.

Yogyakarta, 17 March 2025

Author

TABLE OF CONTENTS

LIST OF FIGURE

CHAPTER I
INTRODUCTION

1.1 Background

Cloud computing has become a critical element in modern IT infrastructure, providing scalable and flexible resources through various service models, including Infrastructure as a Service (IaaS). IaaS enables users to rent virtualized computing resources over the internet, offering flexibility in deploying, managing, and scaling applications without significant hardware investment. Virtual machines (VMs) are integral components of IaaS, serving as isolated environments where software and applications can be installed and executed. This capability is particularly beneficial for web application deployment, where developers and administrators require a controlled and customizable environment.

Deploying a simple web application on a VM typically involves creating and configuring the VM, installing the necessary web server software, and uploading the application files. Through SSH (Secure Shell), users can remotely connect to the VM to execute commands, transfer files, and manage configurations securely. Using SSH ensures encrypted communication, minimizing security risks during the deployment process. The flexibility of VMs allows users to test, modify, and deploy simple web applications without affecting physical hardware or incurring excessive costs.

Google Cloud Platform (GCP) provides a popular solution for creating and managing VM instances through its Compute Engine. GCP offers a user-friendly interface to set up VM instances with customizable configurations, including the choice of operating system, machine type, and network settings. GCP also allows users to connect to these VM instances using SSH directly through the web console or through third-party SSH clients, making the deployment of web applications accessible even to beginners. Additionally, Google Cloud configures firewall rules by default to enhance security, while still providing the flexibility to adjust settings based on project needs.

This approach is popular for small-scale projects, academic purposes, and initial testing before scaling up to more complex architectures. While deploying a

simple web application on a VM through SSH is generally straightforward, challenges can arise related to network configurations, firewall rules, and user access controls. Addressing these challenges effectively requires a clear understanding of the deployment steps and the ability to troubleshoot common issues.

1.2 Problem Formulation

1. How can a simple web application be deployed on a VM through SSH and Linux commands effectively while minimizing configuration errors and security risks?
2. What are the steps required to deploy a simple web application using a VM?
3. How can a secure connection to the VM be established using SSH?
4. How can Google Cloud VM instances be effectively utilized to deploy simple web applications?

1.3 Objectives

1. To demonstrate the use of basic Linux commands for VM management and web server configuration.
2. To provide guidance on securely connecting to VMs via SSH.
3. To illustrate the installation and configuration of a web server (Apache) on the VM.
4. To develop a step-by-step guide for deploying a simple web application on a VM using SSH and Linux.
5. To explore the use of Google Cloud VM instances for simplified web deployment.

1.4 Benefits

1. Users can efficiently deploy web applications on VM instances effectively, minimizing errors and reducing deployment time.
2. Users can implement secure SSH connections to protect sensitive data and reduce security risks during remote management.

3. Users gain hands-on Linux skills on configuring web servers (e.g., Apache) equips users with valuable system administration skills.
4. Users can scale projects using Google Cloud
5. Users can reduce deployment costs because leveraging cloud-based VMs like Google Cloud minimizes the need for physical hardware, lowering deployment expenses.

CHAPTER II

LITERATURE REVIEW

2.1 Infrastructure as a Service

IaaS is a cloud computing model that provides virtualized computing resources over the internet, including virtual machines, storage, and networking components. Unlike traditional infrastructure, IaaS eliminates the need for physical hardware, enabling organizations to scale resources on demand while minimizing operational costs. Bhardwaj et al emphasize the flexibility and cost-effectiveness of IaaS, particularly for deploying web applications. Organizations benefit from the scalability of IaaS, which allows them to adjust resources based on workload demands, ensuring efficient resource utilization without significant capital investment.

IaaS platforms like Google Cloud offer managed VM instances that facilitate easy deployment and management of applications in isolated environments. The ability to create, configure, and manage VMs through a centralized platform streamlines the deployment process for developers. Additionally, IaaS providers handle underlying hardware maintenance and security, allowing users to focus on application development and configuration. As a result, IaaS has become a preferred choice for businesses and individuals looking to deploy web applications quickly and reliably.

2.2 Google Cloud VM Instances

Google Cloud VM instances, provided through Google Cloud Compute Engine, are virtualized servers that can be configured to host various applications. These instances support multiple operating systems and can be accessed via SSH for remote management. Google Cloud's integration of firewall rules and network settings simplifies secure access to VM instances while maintaining flexibility for customization. The platform's user-friendly interface enables users to select machine types, disk sizes, and network configurations, making it accessible for beginners and experienced developers alike.

Mao and Humphrey highlight the efficiency of VM startup time in the cloud, making Google Cloud a practical solution for rapid deployment. The ability to create snapshots and backups of VM instances also enhances data protection and recovery. By utilizing Google Cloud VM instances, developers can deploy simple web applications using Apache, test configurations, and scale their projects with minimal effort. The platform's pay-as-you-go pricing model also makes it cost-effective for small-scale projects and initial testing phases.

2.3 SSH Configuration

SSH (Secure Shell) is a protocol used to establish secure, encrypted connections between a client and a server. In the context of VM deployment, SSH allows users to manage VM instances remotely, transfer files securely, and execute commands. SSH ensures data privacy by encrypting communication between the user's system and the VM, reducing the risk of unauthorized access. Shotts discusses essential Linux commands to configure SSH, such as generating SSH keys (`ssh-keygen`), copying public keys to remote hosts (`ssh-copy-id`), and configuring SSH settings in the `/etc/ssh/sshd_config` file.

Effective SSH configuration is crucial for secure access to VM instances. It is recommended to use strong, unique SSH keys and disable password authentication to enhance security. Additionally, managing firewall rules to restrict SSH access to trusted IP addresses helps prevent unauthorized login attempts. Proper SSH configuration not only secures VM access but also streamlines remote management, making it easier to deploy, monitor, and troubleshoot web applications on VMs.

2.4 Apache Web Server

Apache is a widely used, open-source web server that can be installed on VM instances to host web applications. It is favored for its stability, flexibility, and compatibility with various operating systems, including Linux. Configuring Apache on a VM typically involves installing the package (`sudo apt install apache2`), starting the service (`sudo systemctl start apache2`), and adjusting

firewall rules to allow HTTP/HTTPS traffic. Apache's modular structure also enables users to extend its functionality through plugins and modules.

Deploying a simple web application using Apache on a VM involves creating a dedicated directory for web files, configuring virtual hosts, and managing permissions to ensure accessibility. Google Cloud VM instances provide a suitable environment for this deployment process, with options to configure firewall rules and networking to support Apache's web traffic. Apache's detailed documentation and strong community support further simplify the deployment process, making it accessible even for beginners.

2.5 Basic Linux Commands for Configuring SSH

To establish and manage SSH connections on a Linux-based VM, basic commands are necessary. These include creating and managing SSH keys (`ssh-keygen`), connecting to a remote VM (`ssh user@ip_address`), and transferring files securely (`scp file user@ip_address:/destination`). Proper configuration of the SSH daemon (`sshd_config`) is crucial to ensure secure access. Enderson provides a comprehensive guide on using these Linux commands effectively.

Basic Linux commands also include managing permissions (`chmod`), viewing network configurations (`ifconfig` or `ip addr`), and editing configuration files using text editors like `nano` or `vi`. These skills are essential for setting up SSH securely and configuring Apache on VM instances. Understanding these commands helps users efficiently deploy, manage, and troubleshoot web applications hosted on Google Cloud VM instances.

CHAPTER III
METHODOLOGY

3.1 Problem Analysis

This practical assignment aims to deploy a simple web application on a Google Cloud VM instance by following three main steps: creating a VM instance, securely connecting via SSH, and deploying the Apache web server. The process begins with setting up the VM instance by selecting the appropriate machine type, configuring the operating system, and setting up network rules. Secure access through SSH is established to control the VM remotely, followed by installing and configuring Apache to host the web application. Key challenges include ensuring proper SSH key management, secure access configurations, and accurate web server setup. The expected result is a functional and accessible web application deployed on the cloud, demonstrating an understanding of cloud virtualization, secure remote management, and basic web server configuration.

3.2 Solution Design

3.2.1 Clone Repository Backend

1. Go to Link https://github.com/plirapli/asisten-tcc-2-api-be.git

Figure 1 Step 3.2.1.1

2. Create a database in localhost that suitable in that code

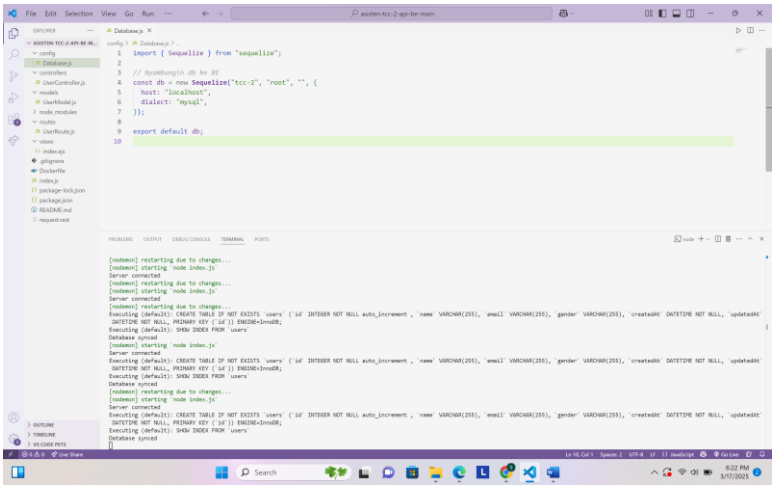


Figure 2 Step 3.2.1.2

3. Run the code and send request to testing the database is it connect with the code or not

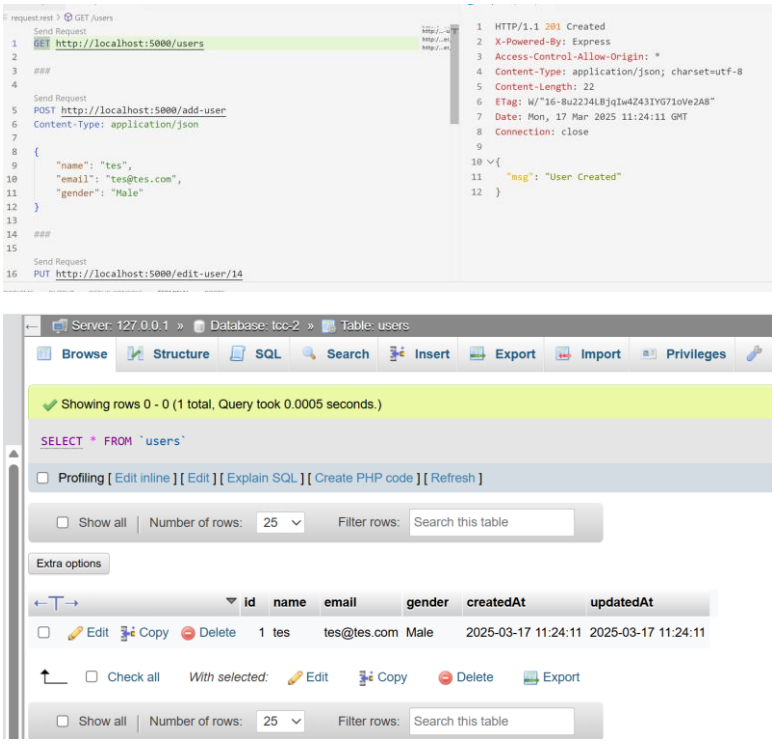


Figure 3 Step 3.2.1.3

3.2.2 Create Instance VM Google Cloud

1. Go to Google Cloud and Select a Project

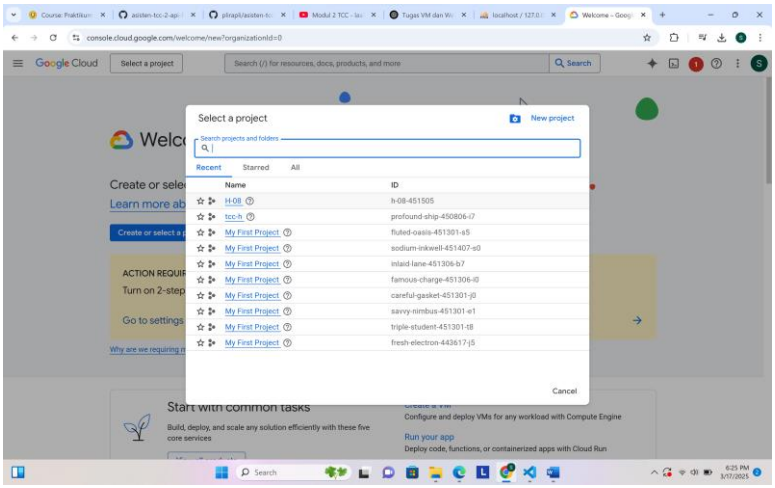


Figure 4 Step 3.2.2.1

2. Choose Compute Engine, and VM Instances

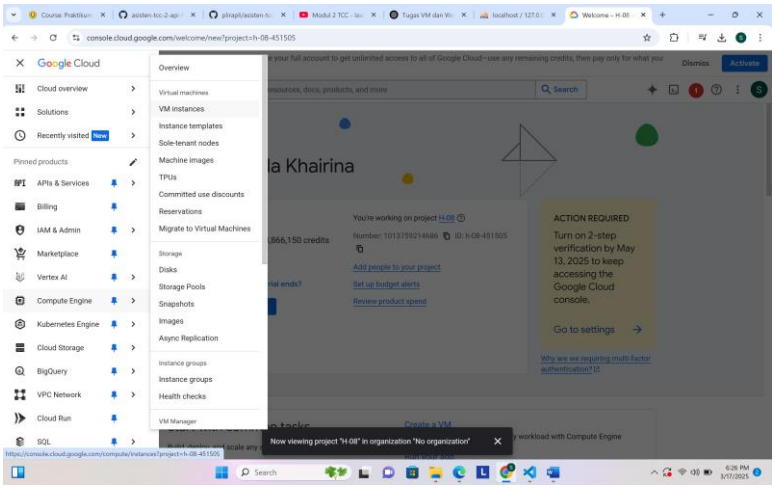


Figure 5 Step 3.2.2.2

3. Select Create VM Instance to make new instance

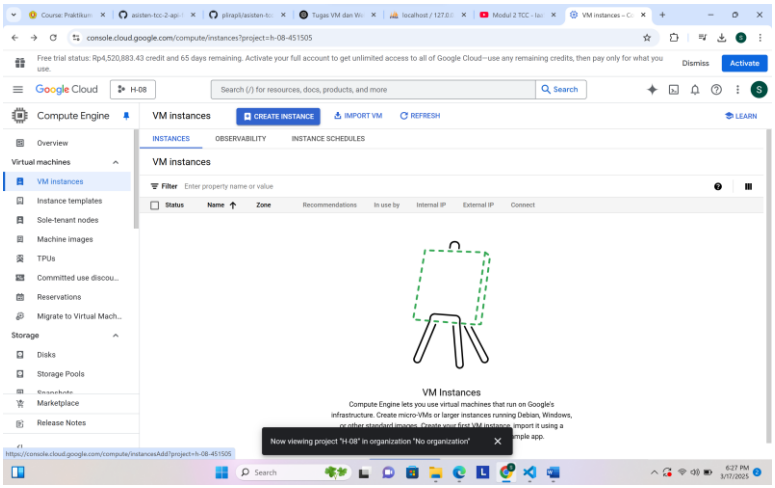


Figure 6 Step 3.2.2.3

4. Configure all the settings and Click Create

E2	Low cost, day-to-day computing	0.25 - 32	1 - 128 GB	Intel Broadwell
N2	Balanced price & performance	2 - 128	2 - 864 GB	Intel Cascade Lake
N2D	Balanced price & performance	2 - 224	2 - 896 GB	AMD Milan
T2A	Scale-out workloads	1 - 48	4 - 192 GB	Ampere Altra
T2D	Scale-out workloads	1 - 60	4 - 240 GB	AMD Milan
N1	Balanced price & performance	0.25 - 96	0.6 - 624 GB	Intel Haswell

Machine type

Choose a machine type with preset amounts of vCPUs and memory that suit most workloads. Or, you can create a custom machine for your workload's particular needs. Learn more

PRESET CUSTOM

g1-small (1 vCPU, 1.7 GB memory)

vCPU	Memory
0.5-1 vCPU (1 shared core)	1.7 GB

Networking

Firewall

Add tags and firewall rules to allow specific network traffic from the Internet

- [x] Allow HTTP traffic
- [x] Allow HTTPS traffic
- [ ] Allow Load Balancer Health Checks

Network tags

Hostname

Set a custom hostname for this instance or leave it default. Choice is permanent

IP forwarding

- [ ] Enable

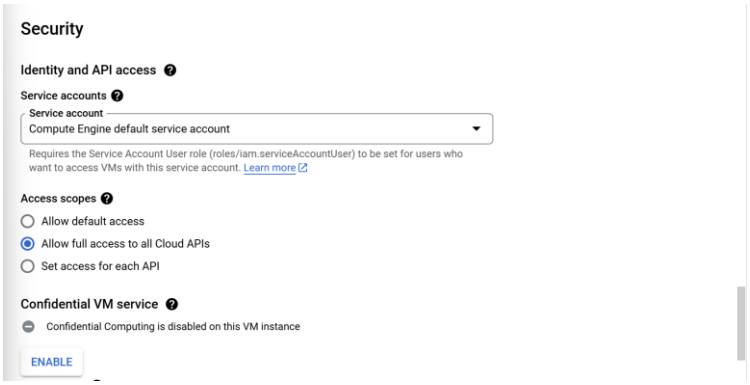


Figure 7 Step 3.2.2.4

5. New Instance will be displayed

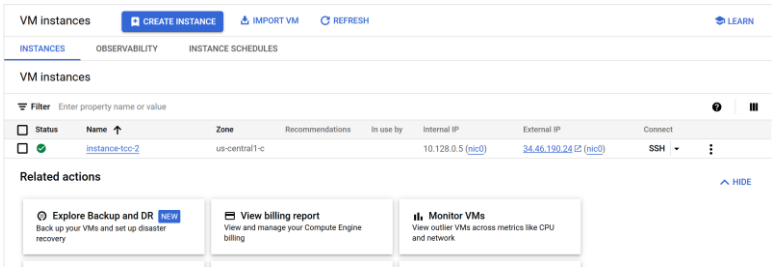


Figure 8 Step 3.2.2.5

3.2.3 Connect Terminal to SSH

1. Open SSH Google Cloud and Authorize

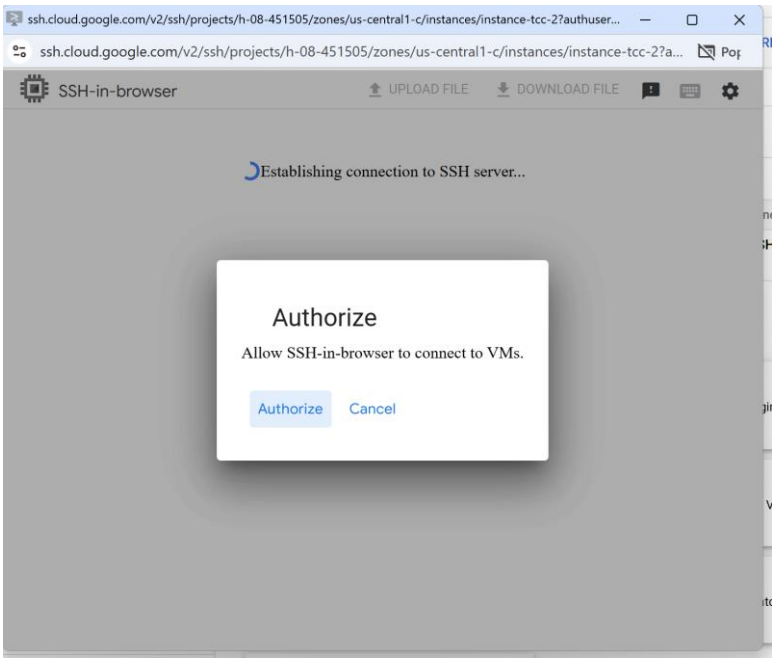


Figure 9 Step 3.2.3.1

2. Go to Link https://cloud.google.com/compute/docs/connect/create-ssh-keys
3. Fill the prompt based on the rules

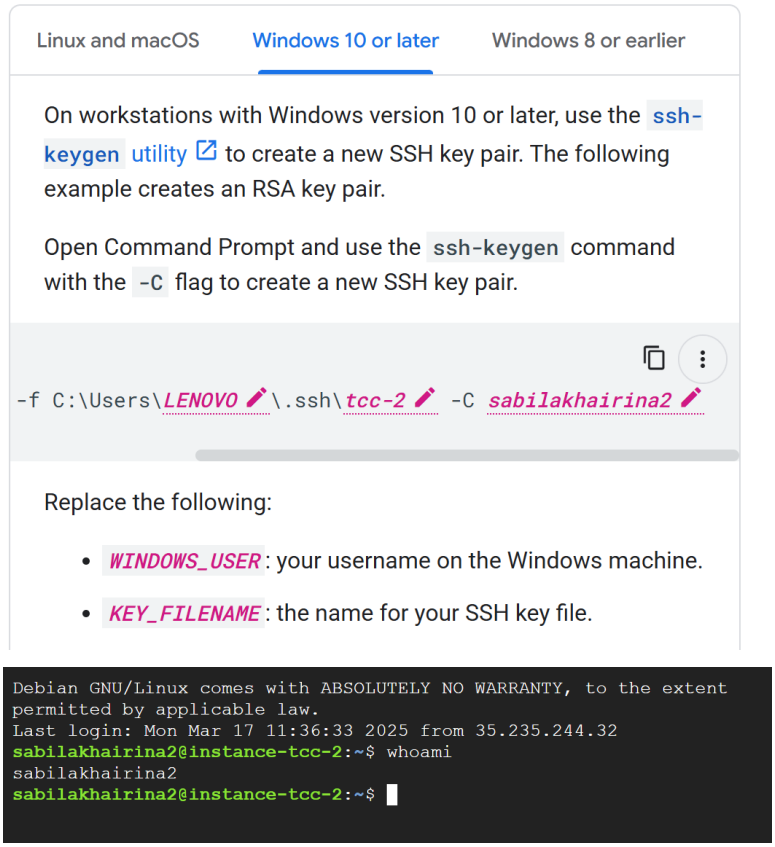


Figure 10 Step 3.2.3.2

4. Copy the prompt to command prompt windows. Make sure that folder .ssh is already in program files

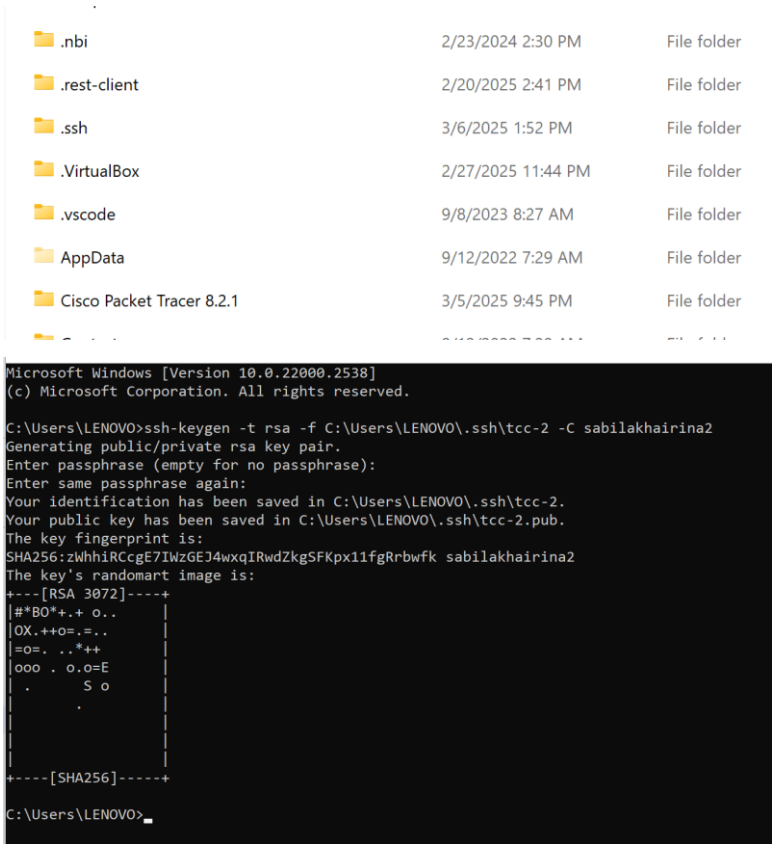


Figure 11 Step 3.2.3.3

5. In command prompt, change directory to folder .ssh and open file pub

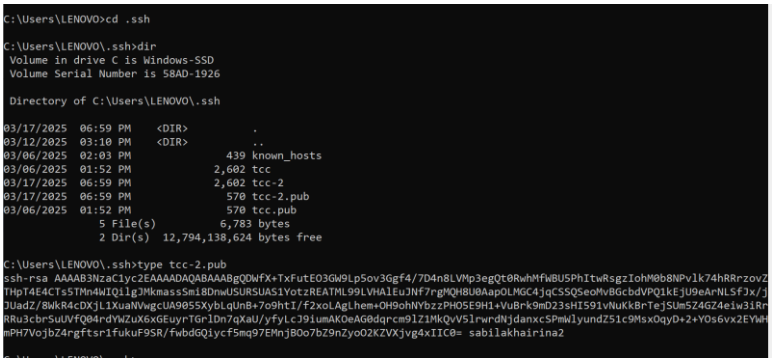


Figure 12 Step 3.2.3.4

6. Copy all the code and paste in SSH key instance. Add item for the ssh key and paste at SSH key 3

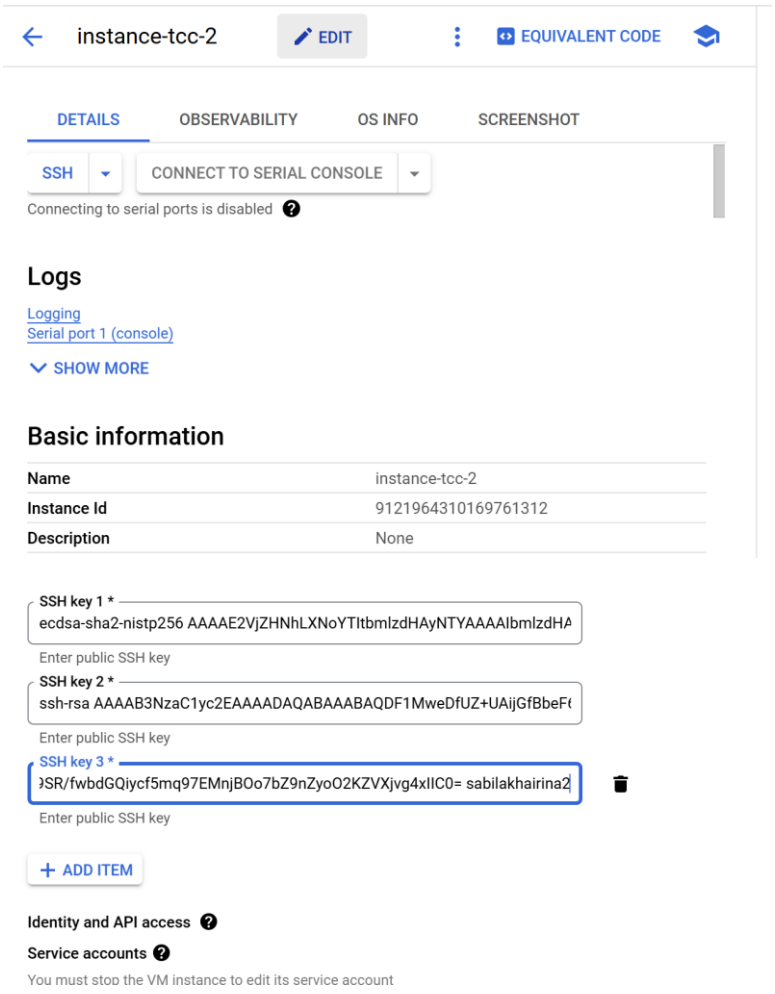


Figure 13 Step 3.2.3.5

3.2.4 Connect to VM

1. Go to Link https://cloud.google.com/compute/docs/connect/standard-ssh#openssh-client

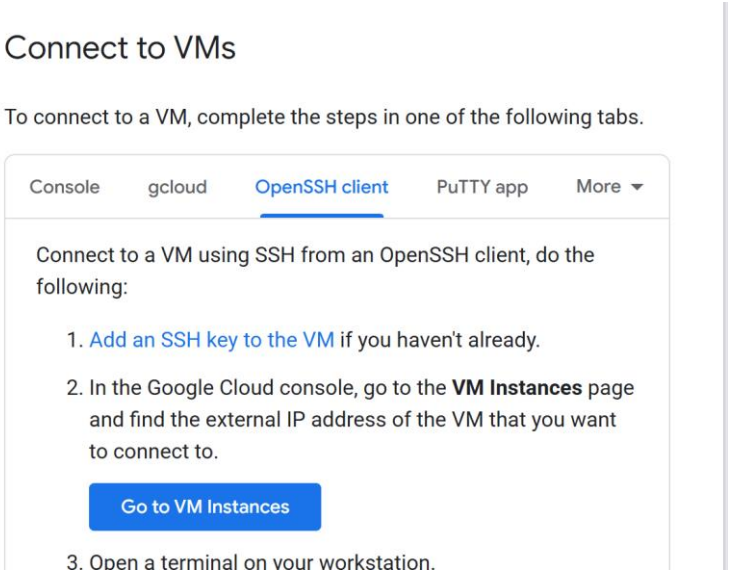


Figure 14 Step 3.2.4.1

2. Config the prompt based on instance external IP

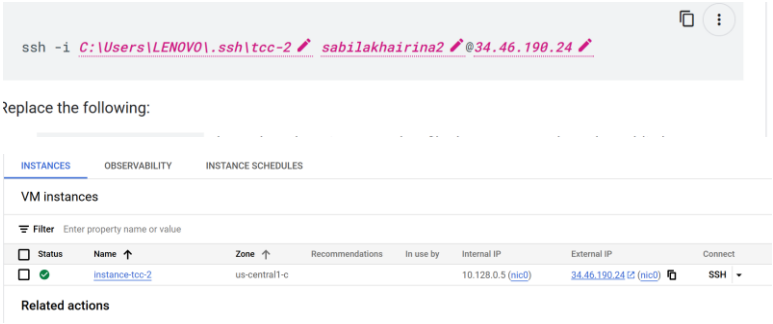


Figure 15 Step 3.2.4.2

3. Copy the prompt and paste to Command Prompt Windows

Figure 16 Step 3.2.4.3

4. Check that ssh and terminal is already connected

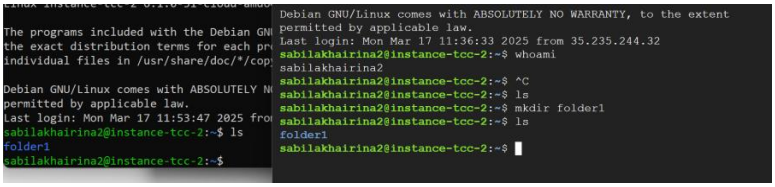
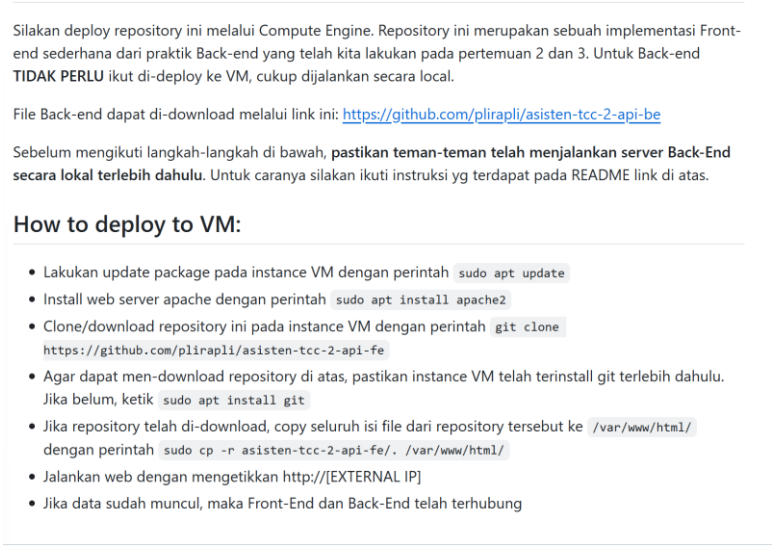


Figure 17 Step 3.2.4.4

3.2.5 Deploy Frontend Web using SSH

1. Go to Link https://github.com/plirapli/asisten-tcc-2-api-fe.git



Silakan deploy repository ini melalui Compute Engine. Repository ini merupakan sebuah implementasi Front-end sederhana dari praktik Back-end yang telah kita lakukan pada pertemuan 2 dan 3. Untuk Back-end **TIDAK PERLU** ikut di-deploy ke VM, cukup dijalankan secara local.

File Back-end dapat di-download melalui link ini: https://github.com/plirapli/asisten-tcc-2-api-be

Sebelum mengikuti langkah-langkah di bawah, **pastikan teman-teman telah menjalankan server Back-End secara lokal terlebih dahulu**. Untuk caranya silakan ikuti instruksi yg terdapat pada README link di atas.

How to deploy to VM:

- Lakukan update package pada instance VM dengan perintah `sudo apt update`
- Install web server apache dengan perintah `sudo apt install apache2`
- Clone/download repository ini pada instance VM dengan perintah `git clone https://github.com/plirapli/asisten-tcc-2-api-fe`
- Agar dapat men-download repository di atas, pastikan instance VM telah terinstall git terlebih dahulu. Jika belum, ketik `sudo apt install git`
- Jika repository telah di-download, copy seluruh isi file dari repository tersebut ke `/var/www/html/` dengan perintah `sudo cp -r asisten-tcc-2-api-fe/. /var/www/html/`
- Jalankan web dengan mengetikkan http://[EXTERNAL IP]
- Jika data sudah muncul, maka Front-End dan Back-End telah terhubung

Figure 18 Step 3.2.5.1

2. Use Basic Linux Commands to clone the repository

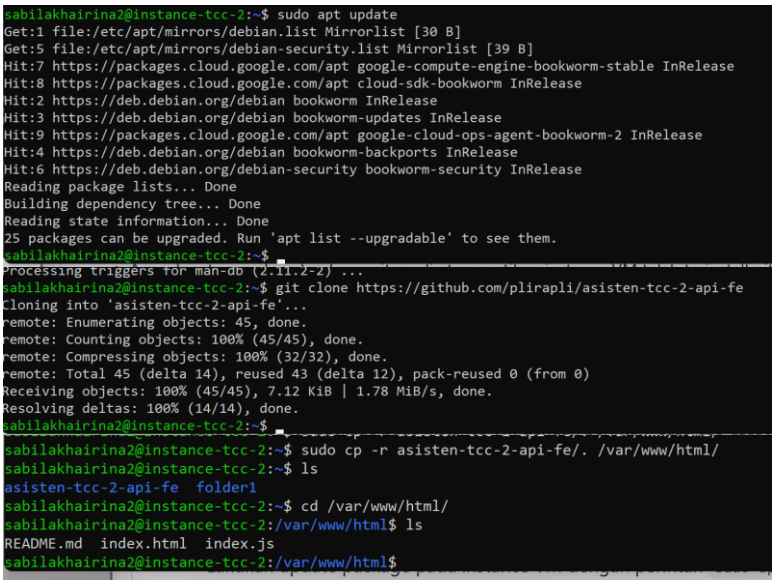


Figure 19 Step 3.2.5.2

3. If cloning repository is done, open the website using external IP instance and replace https with http

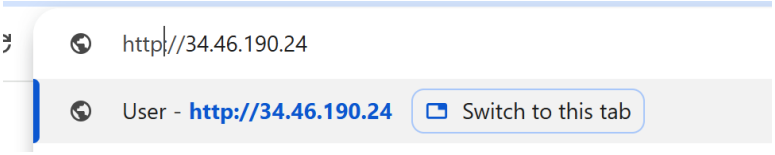


Figure 20 Step 3.2.5.3

4. Fill port based in the code

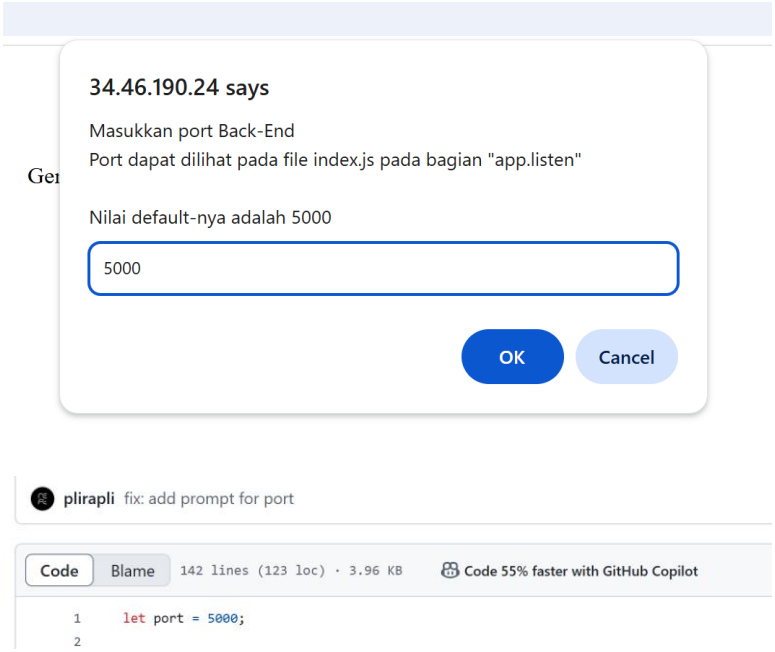
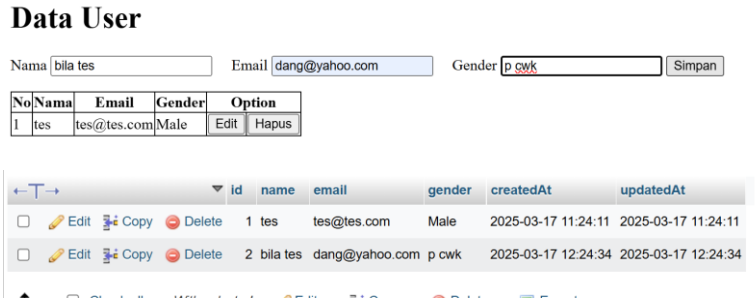


Figure 21 Step 3.2.5.4

5. Try Add, Edit, and Delete data using the website and make sure that front end and back end is already connected



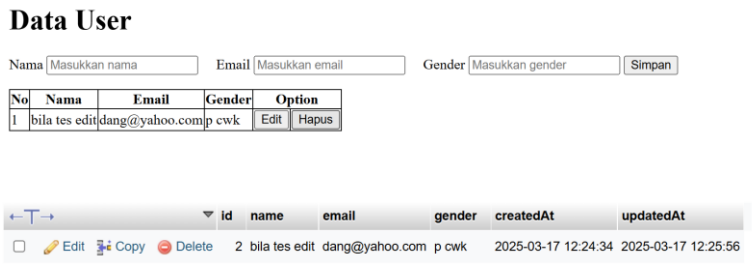


Figure 22 Step 3.2.5.5

CHAPTER IV
RESULTS AND DISCUSSION

4.1 Results

4.1.1 Create Instance VM

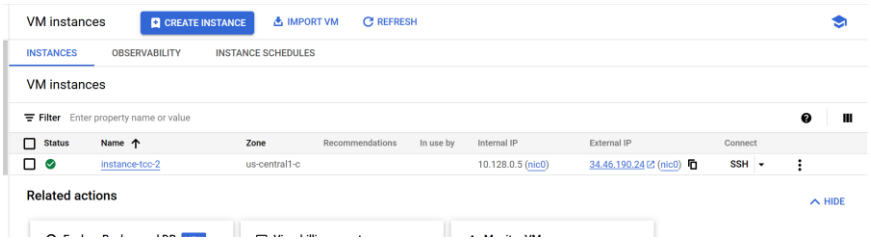


Figure 23 Create Instance VM

The image shows the successful creation of a VM instance named "instance-tcc-2" on Google Cloud, as evidenced by its active status marked with a green check icon. The instance is deployed in the us-central1-c zone, with assigned internal and external IP addresses, making it accessible for SSH connections. The visible SSH option indicates that secure remote access can be configured, allowing users to manage and deploy applications on the VM instance. This step is crucial for subsequent stages, such as setting up a web server using Apache and deploying a simple web application.

4.1.2 Connect to VM Via SSH

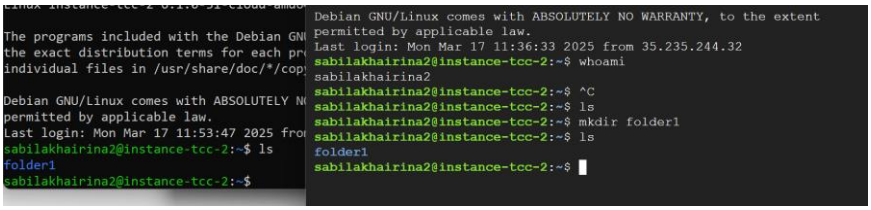


Figure 24 Connect to VM Via SSH

The image demonstrates a successful SSH connection to the VM instance on Google Cloud, evidenced by the command-line interface showing user access and directory manipulation commands. This confirms that the VM is accessible remotely and securely via SSH.

4.1.3 Deploy Website Using Apache on VM

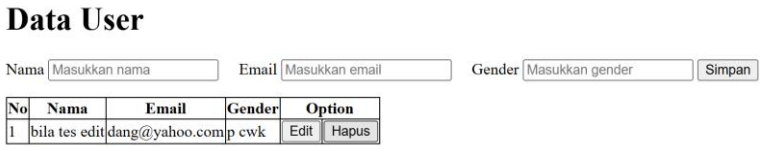


Figure 25 Deploy Website Using Apache on VM

The image shows the successful deployment of a simple web application using Apache on the VM. The displayed web form for adding, editing, and deleting user data validates that the web server is properly configured, connected to the backend, and accessible through the VM's external IP. These outcomes reflect the effective completion of the VM setup, secure connection, and web deployment processes.

4.2 Discussion

The deployment process was structured to guide users through creating a VM instance, securely connecting to it via SSH, and setting up a web server using Apache. The initial step involved cloning the backend repository and preparing a database, ensuring that the environment was properly configured. The second step required creating a VM instance on Google Cloud, configuring machine settings, and establishing necessary firewall rules to enable web traffic. Following this, a secure SSH connection was established, adhering to best practices for key management and access control. Several key configurations, such as correctly setting up SSH keys and managing firewall rules, were crucial to the secure and efficient management of the VM.

Deploying the frontend involved utilizing Linux commands to clone the repository, configuring the Apache server, and testing connectivity. Additionally, configuring Apache’s virtual hosts and permissions ensured the application was accessible via the VM’s external IP address. The step-by-step approach facilitated an organized deployment process, minimizing errors and reinforcing practical skills in server management and network configuration. Despite facing challenges in SSH configuration, network restrictions, and setting correct file permissions for Apache, the deployment was successfully completed, showcasing the potential of cloud-based virtualization for simple web hosting. This comprehensive process addresses the

issues outlined in the problem analysis, demonstrating the feasibility of deploying simple web applications in a virtualized environment.

CHAPTER V
CLOSING

5.1 Conclusion

The deployment of a simple web application using a Google Cloud VM instance, SSH, and Apache demonstrated the practical application of cloud infrastructure and secure remote access techniques. The use of SSH ensured secure communication, while Apache provided a reliable web server environment. The successful completion of this deployment validates the effectiveness of Google Cloud VM instances for web hosting and highlights the significance of proper configuration in ensuring security and accessibility.

.

5.2 Suggestions

To enhance future deployments, it is recommended to further explore automated deployment tools like Ansible or Terraform to streamline the VM provisioning and configuration process. Additionally, integrating SSL/TLS certificates for HTTPS connections can improve security. Continuous monitoring tools should also be considered to ensure server health and uptime, providing a more robust and reliable web hosting solution.

BIBLIOGRAPHY

[1] Manvi, S. S., & Gopal Krishna Shyam. (2013). Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey. *Journal of Network and Computer Applications*, *41*, 424–440. https://doi.org/10.1016/j.jnca.2013.10.004

[2] Bhardwaj, S., Jain, L., & Jain, S. (n.d.). CLOUD COMPUTING: A STUDY OF INFRASTRUCTURE AS A SERVICE (IAAS). *I Nt Ernat Ional Journal of Engineering and I Nform at Ion Technology*, *2*(1), 60–63. Retrieved March 18, 2025, from https://d1wqtxts1xzle7.cloudfront.net/7299777/cloud_computing_a_study_of-libre.pdf?1390850824=&response-content-disposition=inline%3B+filename%3DCloud_computing_A_study_of_infrastructur.pdf&Expires=1742278184&Signature=KhfZg~JTxne75yLkflv9O9cFEHyxG9ZJaO1~bLEqsN3lzeUiM5vw1celNiOcWhU3N8T8sRFM36ONaaiHmt3uPujWjIDzc0Jx~oRqIIUs7KfwHhsq3Qt9VPeHhHTvDC9mWL0GQeEzOkNpb9DTGdU0asm~ekf4s~jQBFj6YvWiy1MCOEGXpgWOg~vCvDGNQDOjNriE0HqC7ySnRXwm9fGz5VIsHnMcmcRtsb8M7~l-hOSuomsCRmarT0oeStilkHlWNZofcwSuq8V80WVVnFLuQ8ly-AO3a-4yl~Unr25jgWusVyL0p-8WHIytO3V~FdpmERDL0w1544LQ4vRBo8csgg__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

[3] Shotts, W. (n.d.). *The Linux Command Line Fourth Internet Edition*. Retrieved March 18, 2025, from https://biblioteca.unisced.edu.mz/bitstream/123456789/1646/1/TLCL.pdf

[4] Mao, M., & Humphrey, M. (2012). A Performance Study on the VM Startup Time in the Cloud. *2012 IEEE Fifth International Conference on Cloud Computing*. https://doi.org/10.1109/cloud.2012.103

[5] IBM. (2024, December 18). *IaaS*. Ibm.com. https://www.ibm.com/think/topics/iaas

[6] Enderson, G. (2025). *Basic Linux Commands*. Wm.edu. https://support.cs.wm.edu/index.php/tips-and-tricks/basic-linux-commands

[7] *Connect to Linux VMs*. (2016). Google Cloud. https://cloud.google.com/compute/docs/connect/standard-ssh#openssh-client

APPENDICES

1. https://github.com/stroberinanas/123220085-TCCPract.git