

# Laporan Cloud Computing Tugas 1

Ricardo Supriyanto / 5025221218

## I. Pendahuluan

Laporan ini bertujuan untuk mendokumentasikan implementasi penggunaan OpenTofu dan Ansible dalam pembangunan infrastruktur untuk sebuah sistem berbasis web yang melibatkan dua komponen utama: web server dan database server. Infrastruktur ini dirancang untuk menjalankan aplikasi berbasis web yang memanfaatkan Apache2 sebagai web server dan MySQL sebagai database server. Web server akan berfungsi untuk melayani permintaan HTTP, sedangkan database server bertugas untuk menyimpan dan mengelola data aplikasi.

Proses penyediaan infrastruktur dimulai dengan penggunaan OpenTofu untuk melakukan provisioning dua *virtual machine* (VM), yaitu **vmWeb** untuk web server dan **vmDB** untuk database server. Setelah itu, Ansible digunakan untuk melakukan konfigurasi lebih lanjut pada kedua VM, termasuk instalasi perangkat lunak yang diperlukan dan pengaturan sistem. Konfigurasi ini mencakup instalasi Apache2 dan repositori github pada web server serta MySQL dan phpMyAdmin pada database server.

Dengan menggunakan pendekatan otomatisasi ini, diharapkan dapat tercipta infrastruktur yang mudah dikelola, *scalable*, dan konsisten di seluruh proses pengembangan dan pengoperasian aplikasi. Laporan ini akan memberikan gambaran rinci mengenai langkah-langkah yang diambil dalam penyediaan infrastruktur menggunakan **OpenTofu** dan **Ansible**, serta penjelasan teknis terkait implementasi yang telah dilakukan.

## II. Deskripsi Tugas

Tujuan Tugas:

1. Mencari dan memilih open source tools untuk cloud orchestration berbasis QEMU/KVM.
2. Menggunakan paradigma Infrastructure as Code (IaC).

Alat yang Digunakan:

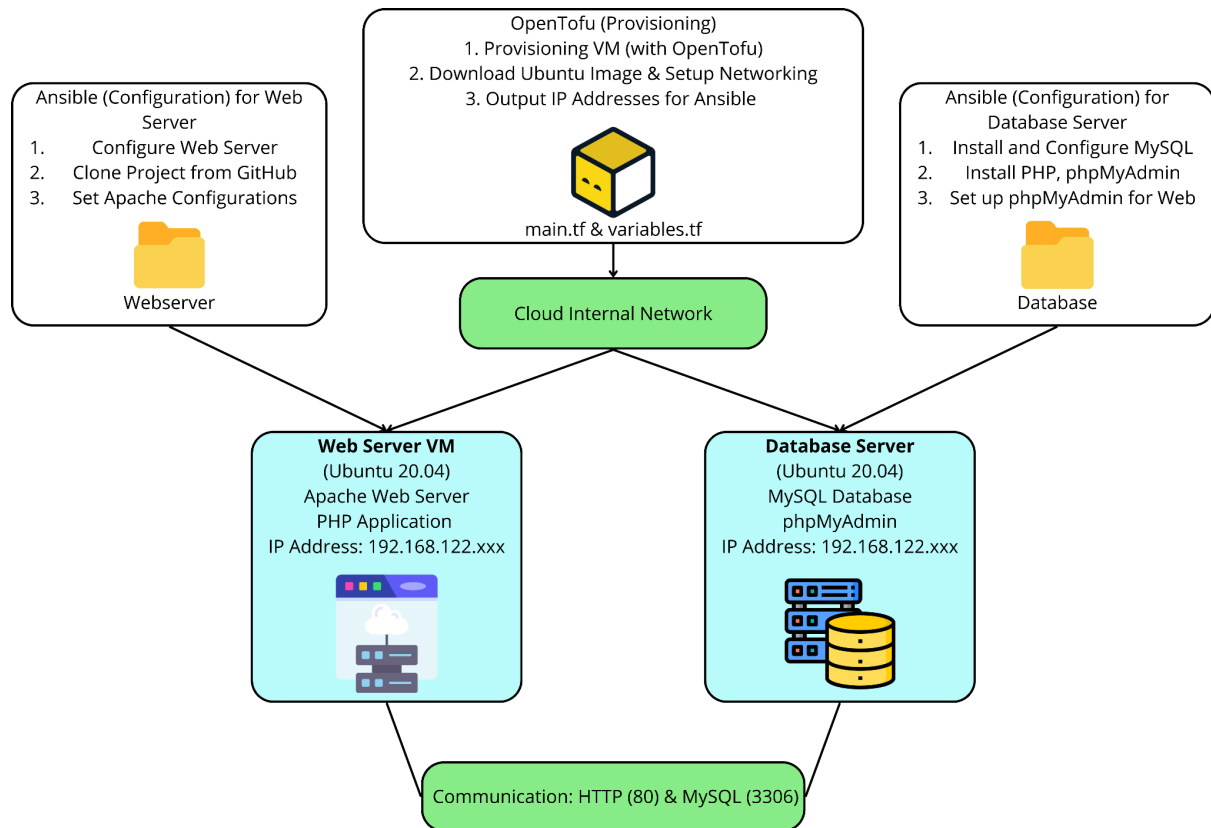
1. Ansible: Alat untuk automasi konfigurasi dan manajemen sistem.
2. OpenTofu: Alternatif open source untuk penyediaan infrastruktur.

Skenario Implementasi:

1. Website Berbasis Apache: Provisi sistem yang menjalankan website menggunakan Apache Web Server, diambil dari repositori GitHub (<https://github.com/rm77/web-sample-6.git>).
2. Database MySQL: Setup server MySQL beserta aplikasi pengelolaannya berbasis web (phpMyAdmin).

### III. Arsitektur dan Implementasi Script

Pada pengerjaan ini terdapat arsitektur sistem ini terdiri dari dua *Virtual Machines* (VM) yang disediakan oleh **OpenTofu** menggunakan Terraform dengan provider **libvirt**, yang berfungsi untuk mengelola infrastruktur di atas hypervisor QEMU/KVM. VM pertama digunakan sebagai **Web Server** yang menjalankan aplikasi web, dan VM kedua digunakan sebagai **Database Server** yang menjalankan MySQL dan phpMyAdmin.



Dengan spesifikasi dari VM seperti demikian:

- Memory: 1024 MB
- vCPU: 1
- Disk Size: 5 GB (volume qcow2)
- IP Address: Dinamis, diperoleh melalui DHCP
- Port: Port 80 untuk akses web server dan port 3306 untuk akses database.

Selanjutnya untuk script terpisah menjadi 2 yakni:

#### a. Provisioning VM menggunakan OpenTofu

Pada tahap pertama, OpenTofu digunakan untuk provisioning *virtual machines* (VMs) yang akan menjalankan website dan database. OpenTofu memanfaatkan Terraform untuk mengelola infrastruktur dan menggunakan provider *libvirt* untuk berinteraksi dengan QEMU/KVM. Dalam proyek ini, membuat dua VM: satu untuk web server dan satu untuk database server.

```

1 terraform {
2   required_providers {
3     libvirt = {
4       source = "dmacvilar/libvirt"
5       version = "0.8.1"
6     }
7   }
8 }
9
10 provider "libvirt" {
11   uri = "qemu:///system"
12 }

```

**provider "libvirt"**: Mengkonfigurasi provider untuk menggunakan QEMU/KVM dengan URI `qemu:///system`, yang akan digunakan untuk mengelola *virtual machine* di host lokal.

```

1 resource "null_resource" "cache_image" {
2   provisioner "local-exec" {
3     command = "wget -O /tmp/ubuntu-20.04.qcow2 ${var.ubuntu_image}"
4   }
5 }

```

**resource "null\_resource" "cache\_image"**: Menyediakan resource sementara untuk mendownload image Ubuntu 20.04 dari URL yang ditentukan dan menyimpannya di `/tmp/ubuntu-20.04.qcow2`. Ini diperlukan untuk pembuatan VM.

```

1 resource "libvirt_volume" "base" {
2   name     = "base.qcow2"
3   source   = "/tmp/ubuntu-20.04.qcow2"
4   pool     = var.libvirt_disk_path
5   format   = "qcow2"
6   depends_on = [null_resource.cache_image]
7 }

```

**resource "libvirt\_volume" "base"**: Membuat volume baru dari file image yang sudah diunduh, dengan format `qcow2`. Volume ini menjadi dasar untuk pembuatan disk *virtual machine*.

```

1  resource "libvirt_domain" "ubuntu" {
2      count = length(var.vm)
3      name = var.vm[count.index]
4      memory = "1024"
5      vcpu = 1
6
7      cloudinit = libvirt_cloudinit_disk.commoninit[count.index].id
8
9      network_interface {
10         network_name = "default"
11         wait_for_lease = true
12         hostname = var.vm[count.index]
13     }
14
15     console {
16         type = "pty"
17         target_port = "0"
18         target_type = "serial"
19     }
20
21     console {
22         type = "pty"
23         target_type = "virtio"
24         target_port = "1"
25     }
26
27     disk {
28         volume_id = libvirt_volume.ubuntu20-qcow2[count.index].id
29     }
30
31     graphics {
32         type = "spice"
33         listen_type = "address"
34         autoport = true
35     }
36 }

```

**resource "libvirt\_domain" "ubuntu":** Mengonfigurasi *virtual machine* (VM). Di sini mendefinisikan VM dengan menggunakan template disk yang sudah dibuat sebelumnya, dan mengatur memori, vCPU, serta network interface.

### Output IP Address:

```

1  output "web_server_ip" {
2      value = libvirt_domain.ubuntu[1].network_interface[0].addresses[0]
3  }
4
5  output "db_server_ip" {
6      value = libvirt_domain.ubuntu[0].network_interface[0].addresses[0]
7  }
8

```

Output ini memberikan IP address yang digunakan oleh VM web server dan database. IP ini digunakan pada konfigurasi selanjutnya untuk Ansible.

## b. Konfigurasi Web Server dan Database dengan Ansible

### 1. Konfigurasi Web Server (Apache dan Repositori)

Web server yang digunakan adalah Apache, dan untuk menjalankan aplikasi HTML, juga perlu menginstal HTML dan beberapa modul yang dibutuhkan. Berikut adalah langkah-langkah dan penjelasan dari playbook Ansible untuk konfigurasi web server:

```
1 ---
2 - name: Install Apache2 and Dependencies
3   ansible.builtin apt:
4     name:
5       - apache2
6       - apt-transport-https
7       - ca-certificates
8       - software-properties-common
9     state: present
10    update_cache: true
11
12 - name: Clone Project from GitHub
13   ansible.builtin.git:
14     repo: https://github.com/rm77/web-sample-6.git
15     dest: /var/www/html/web-sample-6
16     clone: true
17     force: yes
18   notify:
19     - Restart Apache
20
21 - name: Add safe directory for Git
22   ansible.builtin.command:
23     cmd: git config --global --add safe.directory /var/www/html/web-sample-6
24   become: true
25   notify:
26     - Restart Apache
27
28 - name: Copy Apache Default Config
29   ansible.builtin.template:
30     src: 000-default.conf.j2
31     dest: /etc/apache2/sites-available/000-default.conf
32     mode: '0644'
33
34 - name: Set Directory Permissions
35   ansible.builtin.file:
36     path: /var/www/html/web-sample-6
37     state: directory
38     recurse: true
39     mode: "775"
40   notify:
41     - Restart Apache
42
43 - name: Set Ownership for Web Directory
44   ansible.builtin.file:
45     path: /var/www/html/web-sample-6
46     owner: www-data
47     group: www-data
48     recurse: true
49     mode: "0775"
50   notify:
51     - Restart Apache
52
```

Script Ansible pertama menginstal **Apache2** dan dependensinya, kemudian mengkloning aplikasi dari **GitHub** ke direktori /var/www/html/web-sample-6. Setelah aplikasi terkloning, Git dikonfigurasi agar dapat beroperasi dengan aman. Konfigurasi Apache diatur menggunakan file template (000-default.conf.j2) untuk menunjuk ke aplikasi yang baru diunduh. Hak akses dan kepemilikan direktori juga diatur, memastikan Apache dapat mengakses dan menyajikan aplikasi web dengan benar.

## 2. Konfigurasi Database Server (MySQL dan phpMyAdmin)

Untuk server database, Ansible digunakan untuk konfigurasi MySQL dan phpMyAdmin, alat berbasis web untuk manajemen MySQL. Berikut adalah langkah-langkah dan penjelasan dari playbook Ansible untuk konfigurasi database server:

```
1  - name: Update system and install prerequisites
2  ansible.builtin.apt:
3      name:
4          - aptitude
5          - mysql-server
6          - mysql-client
7          - python3-mysqldb
8          - apt-transport-https
9          - ca-certificates
10         - software-properties-common
11     state: present
12     update_cache: true
13
14 - name: Create MySQL Database
15 community.mysql.mysql_db:
16     name: "{{ db_name }}"
17     state: present
18
19 - name: Create MySQL User
20 community.mysql.mysql_user:
21     name: "{{ db_user }}"
22     password: "{{ db_pass }}"
23     encrypted: false
24     priv: "{{ db_name }}.*:ALL"
25     state: present
26
27 - name: Allow MySQL remote login
28 ansible.builtin.lineinfile:
29     path: /etc/mysql/mysql.conf.d/mysqld.cnf
30     regexp: '^bind-address'
31     line: 'bind-address = 0.0.0.0'
32     backup: true
33     notify:
34         - Restart MySQL
35
36 - name: Add PHP repository
37 ansible.builtin.apt_repository:
38     repo: "ppa:ondrej/php"
39     state: present
40
41 - name: Install PHP, phpMyAdmin, and dependencies
42 ansible.builtin.apt:
43     name:
44         - php8.0-mysql
45         - php8.0-mbstring
46         - php8.0-xml
47         - php8.0-curl
48         - php8.0-cli
49         - php8.0-common
50         - php8.0-gd
51         - libapache2-mod-php8.0
52         - phpmyadmin
53         - debconf-utils
54     state: present
55     update_cache: true
```

```

1  - name: Configure debconf for phpMyAdmin
2  debconf:
3      name: phpmyadmin
4      question: "{{ item.question }}"
5      value: "{{ item.value }}"
6      vtype: "{{ item.vtype }}"
7  loop:
8      - { question: 'phpmyadmin/dbconfig-install', value: 'true', vtype: 'boolean' }
9      - { question: 'phpmyadmin/app-password-confirm', value: 'password', vtype: 'password' }
10     - { question: 'phpmyadmin/mysql/admin-pass', value: '{{ db_pass }}', vtype: 'password' }
11     - { question: 'phpmyadmin/mysql/app-pass', value: '{{ db_pass }}', vtype: 'password' }
12     - { question: 'phpmyadmin/reconfigure-webserver', value: 'apache2', vtype: 'multiselect' }
13
14 - name: Configure phpMyAdmin Apache
15 ansible.builtin.template:
16     src: phpmyadmin.conf.j2
17     dest: /etc/apache2/conf-available/phpmyadmin.conf
18     owner: rics
19     group: rics
20     mode: '0644'
21
22 - name: Enable phpMyAdmin Apache Configuration
23 ansible.builtin.command:
24     cmd: a2enconf phpmyadmin.conf
25     notify:
26         - Restart Apache
27
28 - name: Enable PHP 8.0 module
29 ansible.builtin.command:
30     cmd: a2enmod php8.0
31
32 - name: Enable Apache rewrite module
33 ansible.builtin.command:
34     cmd: a2enmod rewrite
35
36 - name: Set Mode for phpMyAdmin
37 ansible.builtin.file:
38     path: /usr/share/phpmyadmin
39     state: directory
40     recurse: true
41     mode: "775"
42
43 - name: Set Ownership for phpMyAdmin
44 ansible.builtin.file:
45     path: /usr/share/phpmyadmin
46     owner: www-data
47     group: www-data
48     recurse: true
49     mode: "0775"
50     notify:
51         - Restart Apache

```

Pada server database, script Ansible menginstal **MySQL** beserta dependensinya. Database baru dibuat dengan menggunakan module `mysql_db` dan pengguna baru dibuat dengan hak akses penuh menggunakan module `mysql_user`. Agar MySQL dapat diakses secara remote, konfigurasi `bind-address` diubah. Selanjutnya, **PHP** dan **phpMyAdmin** diinstal untuk mengelola database. Konfigurasi phpMyAdmin dilakukan dengan `debconf` dan file konfigurasi Apache untuk phpMyAdmin disalin menggunakan `template`. Terakhir, **modul PHP** dan **modul rewrite Apache** diaktifkan agar aplikasi dapat berjalan dengan baik.

#### IV. Screenshot Implementasi dan Penjelasannya

Pada bagian ini, menunjukkan screenshot yang diambil selama implementasi dan menjelaskan setiap langkah yang terjadi pada masing-masing.

```
Apply complete! Resources: 8 added, 0 changed, 0 destroyed.

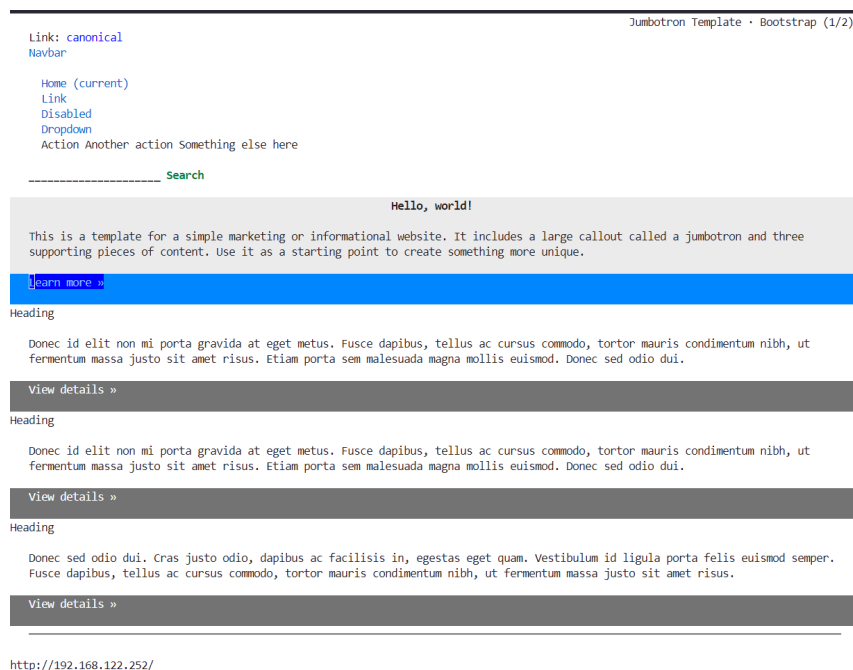
Outputs:

db_server_ip = "192.168.122.158"
web_server_ip = "192.168.122.91"
  Id  Name      State
  ----
  3   mesin-db  running
  4   mesin-web  running
```

- **Penjelasan:** Screenshot ini menunjukkan status eksekusi provisioning VM menggunakan OpenTofu. Di sini, dapat melihat bahwa VM sedang dibuat dan diinstal menggunakan image Ubuntu 20.04.

```
PLAY RECAP *****
192.168.122.158      : ok=16  changed=15  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
192.168.122.91      : ok=8    changed=7  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

- **Penjelasan:** Screenshot ini memperlihatkan proses instalasi menggunakan Ansible untuk Apache pada VM yang berfungsi sebagai web server dan phpMyAdmin pada VM yang berfungsi sebagai database server.



- **Penjelasan:** Setelah instalasi dan konfigurasi berhasil, screenshot ini menunjukkan tampilan halaman web dari aplikasi yang dijalankan pada web server, yang berisi aplikasi yang di-clone dari GitHub.





- **Penjelasan:** Screenshot ini memperlihatkan tampilan phpMyAdmin yang terhubung ke database MySQL. Hal ini menunjukkan bahwa database dan phpMyAdmin berhasil dikonfigurasi dengan benar.

## V. Kesimpulan

**Infrastructure as Code (IaC)** menggunakan **OpenTofu** dan **Ansible** untuk mengelola infrastruktur dan konfigurasi sistem. Dengan OpenTofu, berhasil membuat dan melakukan konfigurasi dua *virtual machines* berbasis QEMU/KVM, masing-masing untuk menjalankan *web server* dan *database server*. Proses ini dilakukan secara otomatis melalui script Terraform yang telah diintegrasikan.

Setelah *provisioning VM* selesai, Ansible digunakan untuk konfigurasi *server web* dan *database server* untuk instalasi Apache, PHP, MySQL, serta phpMyAdmin, sehingga aplikasi web dapat dijalankan dan database dapat dikelola melalui antarmuka web. Pendekatan ini memungkinkan untuk melakukan provisioning dan konfigurasi infrastruktur secara otomatis, terstruktur, dan dapat diulang kapan saja, yang sangat berguna dalam pengelolaan infrastruktur cloud.

### Keunggulan Pendekatan IaC:

- **Kecepatan dan Konsistensi:** Proses provisioning dan konfigurasi dapat dilakukan dengan cepat dan konsisten di berbagai lingkungan tanpa kesalahan manual.
- **Pemeliharaan dan Skalabilitas:** Dengan menggunakan GitHub, setiap perubahan pada konfigurasi dapat dicatat dan dikelola, memungkinkan pemeliharaan dan ekspansi sistem di masa depan.
- **Automasi:** Menggunakan OpenTofu dan Ansible memungkinkan untuk mengotomatisasi seluruh proses setup dan konfigurasi server dengan script yang dapat diulang di masa depan.

Secara keseluruhan, tugas ini menggunakan **OpenTofu** dan **Ansible** untuk mengelola infrastruktur cloud, serta bagaimana mengintegrasikan berbagai komponen konfigurasi untuk membangun *virtual machine* yang dapat digunakan oleh *end-user*.