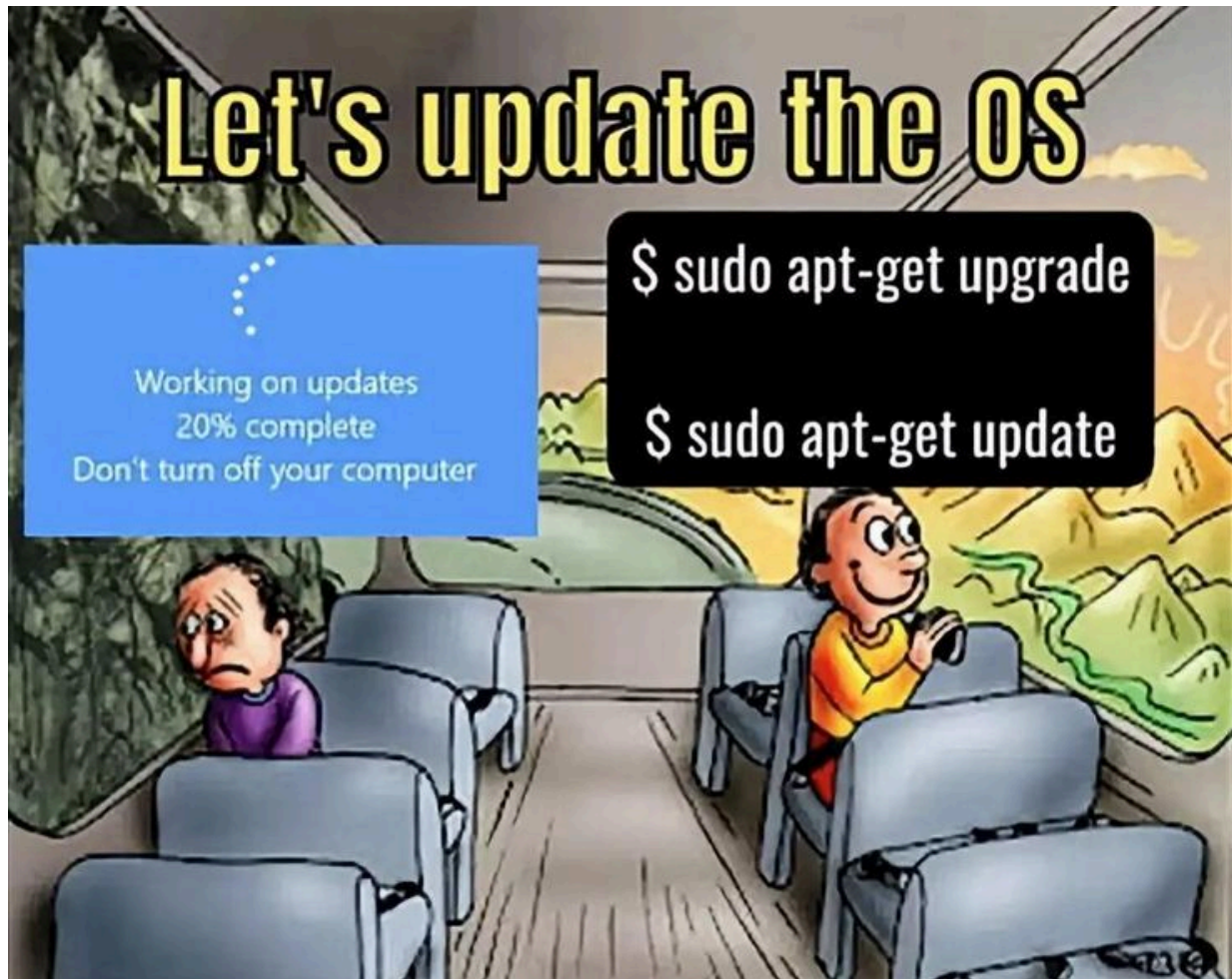


Tugas 3 Ansible

Disusun oleh:

Muhammad Amir Al Aqwa



Pendahuluan

Tugas ini merupakan implementasi dari materi Creating and Managing Ansible Playbooks with Templates and Loops, yang bertujuan untuk menunjukkan kemampuan dalam mengotomatisasi manajemen konfigurasi menggunakan Ansible. Dalam laporan ini, akan disajikan sebuah playbook Ansible yang memanfaatkan teknologi Jinja2 Templating untuk menghasilkan file konfigurasi yang dinamis, serta penggunaan Loop untuk menjalankan serangkaian tugas secara efisien dan berulang. Implementasi ini mencerminkan praktik terbaik dalam Infrastructure as Code (IaC) dengan memisahkan logika (playbook) dari data konfigurasi (template) dan memastikan konsistensi deployment.

Langkah Praktikum

1. Setup Environment

Tugas ini akan mensimulasikan *deployment* konfigurasi web server (Nginx) di beberapa *user* di *Managed Node* Ubuntu.

1. Pastikan control node sudah terkoneksi dengan managed node dan sudah bisa ssh.
2. Menyiapkan direktori kerja baru dengan struktur kerja seperti ini.

```
.
├── inventory.ini
├── playbook.yaml
└── templates
    └── nginx_user.conf.j2
```

3. Kemudian membuat file inventory.ini dengan isi seperti ini (IP host dan username host dapat disesuaikan sesuai dengan machine Anda)

```
1 [webservers]
2 ubuntu_host ansible_host=192.168.100.215 ansible_user=minkwaq
```

2. Creating Jinja template

1. Lalu membuat file template Jinja2 di direktori ./templates dengan isi sebagai berikut

```

1 server {
2     listen {{ nginx_port }};
3     server_name {{ item.user }}.{{ website_domain }};
4     ---
5     root /var/www/{{ item.user }}/html;
6     index index.html;
7     ---
8     location / {
9         try_files $uri $uri/ =404;
10    }
11
12    # Log file khusus
13    access_log /var/log/nginx/{{ item.user }}_access.log;
14    error_log /var/log/nginx/{{ item.user }}_error.log;

```

3. Creating and running the Playbooks

1. Buat file dengan nama `playbook.yaml` yang berisi konfigurasi sebagai berikut

```

42 ---
41 - name: Membuat dan Mengelola Konfigurasi Web Server dengan Loop & Template
40   hosts: webservers
39   become: yes
38
37   vars:
36     website_domain: local
35     nginx_port: 80
34   -
33     users_to_deploy:
32       - user: budi
31         | uid: 1001
30       - user: caca
29         | uid: 1002
28       - user: dani
27         | uid: 1003
26
25   tasks:
24     - name: 1. Pastikan Nginx terinstal
23       ansible.builtin.package:
22         name: nginx
21         state: present
20       tags:
19         - setup
18
17     - name: 2. Implementasi Loop
16       ansible.builtin.user:
15         name: "{{ item.user }}"
14         uid: "{{ item.uid }}"
13         shell: /bin/bash
12         state: present
11         loop: "{{ users_to_deploy }}"
10         loop_control:
9           loop_var: item
8         tags:
7           - users
6
5     - name: 3. Implementasi Loop & Template
4       ansible.builtin.template:
3         src: templates/nginx_user.conf.j2
2         dest: "/etc/nginx/sites-available/{{ item.user }}.conf"
1         owner: root
43        group: root
1         mode: '0644'
2         loop: "{{ users_to_deploy }}"
3         notify: Reload Nginx
4         tags:

```

```

tags:
  - config

- name: 4. Implementasi Loop
  ansible.builtin.file:
    src: "/etc/nginx/sites-available/{{ item.user }}.conf"
    dest: "/etc/nginx/sites-enabled/{{ item.user }}.conf"
    state: link
  loop: "{{ users_to_deploy }}"
  notify: Reload Nginx
  tags:
    - config

- name: 5. Implementasi Loop
  ansible.builtin.file:
    path: "/var/www/{{ item.user }}/html"
    state: directory
    owner: "{{ item.user }}"
    group: "{{ item.user }}"
    mode: '0755'
  loop: "{{ users_to_deploy }}"
  tags:
    - webroot

- name: 6. Membuat dummy index.html (Opsional)
  ansible.builtin.copy:
    content: "<h1>Selamat Datang di Web {{ item.user }}</h1>"
    dest: "/var/www/{{ item.user }}/html/index.html"
    owner: "{{ item.user }}"
    group: "{{ item.user }}"
    mode: '0644'
  loop: "{{ users_to_deploy }}"

handlers:
  - name: Reload Nginx
    ansible.builtin.service:
      name: nginx
      state: reloaded

```

2. Jalankan playbook dengan perintah sebagai berikut dan masukkan password

```

❌ ▶ ansible-playbook -i inventory.ini playbook.yaml --ask-become-pass
BECOME password:

```

3. Berikut output jika berhasil:

Instalasi Nginx (jika belum, jika sudah maka outputnya akan seperti ini)

```
-----  
< TASK [1. Pastikan Nginx terinstal] >  
-----  
      ^__^  
      (oo)\_____  
      (__)\\       )\\/\  
           ||----w |  
           ||     ||  
  
ok: [ubuntu_host]
```

Dibuat 3 user

```

< TASK [2. Implementasi Loop] >
-----
      ^__^
      (oo)\_______
      (__)\\       )\/\
          ||----w |
          ||     ||

changed: [ubuntu_host] => (item={'user': 'budi', 'uid': 1001})
changed: [ubuntu_host] => (item={'user': 'caca', 'uid': 1002})
changed: [ubuntu_host] => (item={'user': 'dani', 'uid': 1003})
-----

< TASK [3. Implementasi Loop & Template] >
-----
      ^__^
      (oo)\_______
      (__)\\       )\/\
          ||----w |
          ||     ||

changed: [ubuntu_host] => (item={'user': 'budi', 'uid': 1001})
changed: [ubuntu_host] => (item={'user': 'caca', 'uid': 1002})
changed: [ubuntu_host] => (item={'user': 'dani', 'uid': 1003})
-----

< TASK [4. Implementasi Loop] >
-----
      ^__^
      (oo)\_______
      (__)\\       )\/\
          ||----w |
          ||     ||

changed: [ubuntu_host] => (item={'user': 'budi', 'uid': 1001})
changed: [ubuntu_host] => (item={'user': 'caca', 'uid': 1002})
changed: [ubuntu_host] => (item={'user': 'dani', 'uid': 1003})
-----

< TASK [5. Implementasi Loop] >
-----
      ^__^
      (oo)\_______
      (__)\\       )\/\
          ||----w |
          ||     ||

changed: [ubuntu_host] => (item={'user': 'budi', 'uid': 1001})
changed: [ubuntu_host] => (item={'user': 'caca', 'uid': 1002})
changed: [ubuntu_host] => (item={'user': 'dani', 'uid': 1003})
-----

```

,

Layanan Nginx akan di reload

```

-----
< RUNNING HANDLER [Reload Nginx] >
-----

      ^__^
      (oo)\_______
      (_____)       )\\/\
              ||----w |
              ||     ||

changed: [ubuntu_host]

```

hasil akhir ok

```

-----
< PLAY RECAP >
-----

      ^__^
      (oo)\_______
      (_____)       )\\/\
              ||----w |
              ||     ||

ubuntu_host      : ok=8    changed=6    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

user dibuat

```

budi:x:1001:1001::/home/budi:/bin/bash
caca:x:1002:1002::/home/caca:/bin/bash
dani:x:1003:1003::/home/dani:/bin/bash

```

Penutup

Implementasi *playbook* Ansible yang telah disajikan dalam laporan ini berhasil memenuhi semua persyaratan tugas dengan menerapkan **Jinja2 Templating** untuk manajemen konfigurasi yang dinamis dan **Loop** untuk optimasi efisiensi tugas. Melalui pemanfaatan kedua fitur utama ini, proses *deployment* dan konfigurasi layanan di beberapa *host* dapat dilakukan secara otomatis, konsisten, dan minim kesalahan manual. Diharapkan hasil implementasi ini dapat memberikan pemahaman yang jelas mengenai kekuatan Ansible dalam orkestrasi dan otomatisasi infrastruktur.