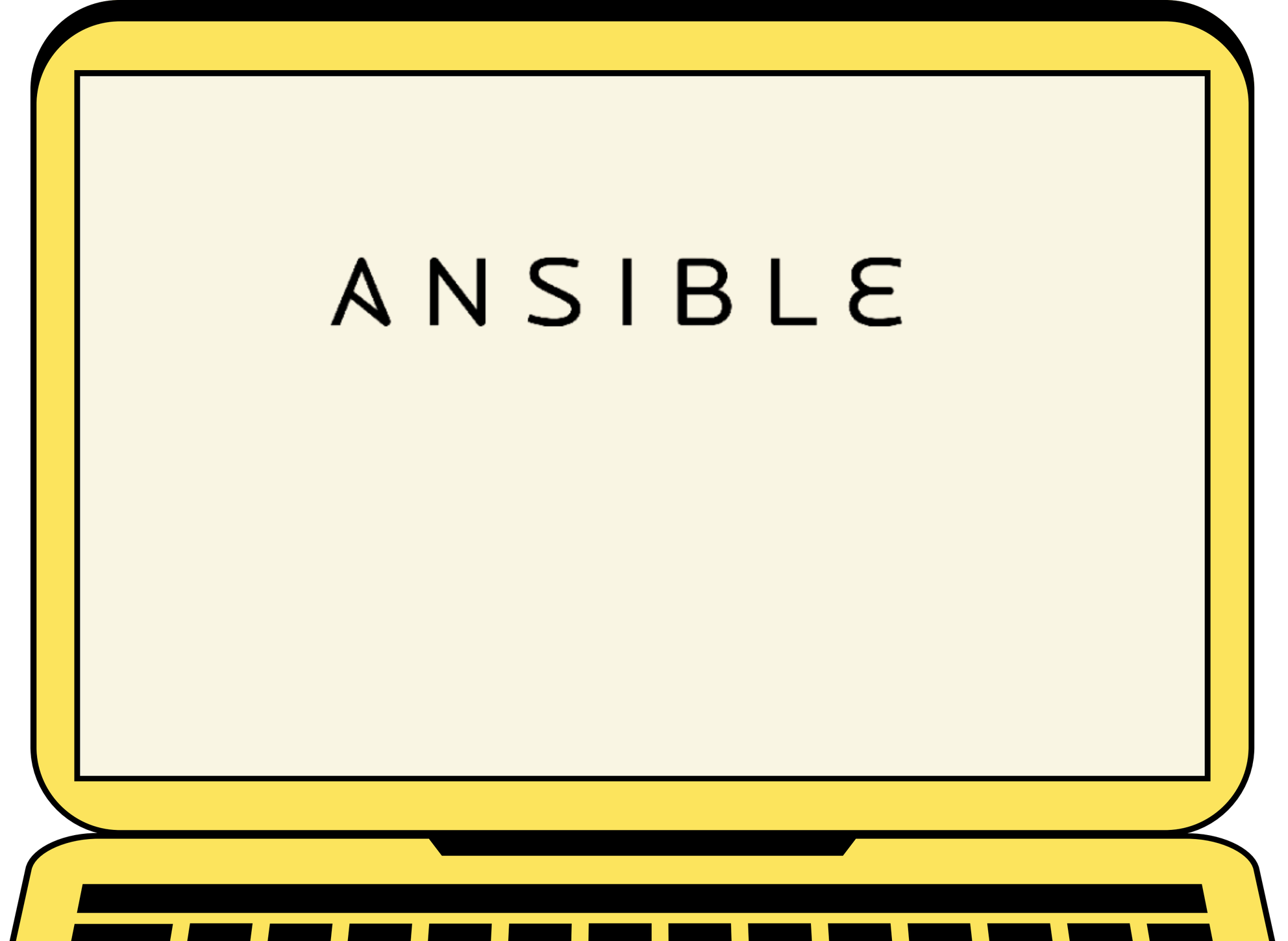

ANSIBLE

Wat is ansible?

Tool voor:

- configuratie management
- (cloud) provisioning
- applicatie deployment

Waarom ansible?

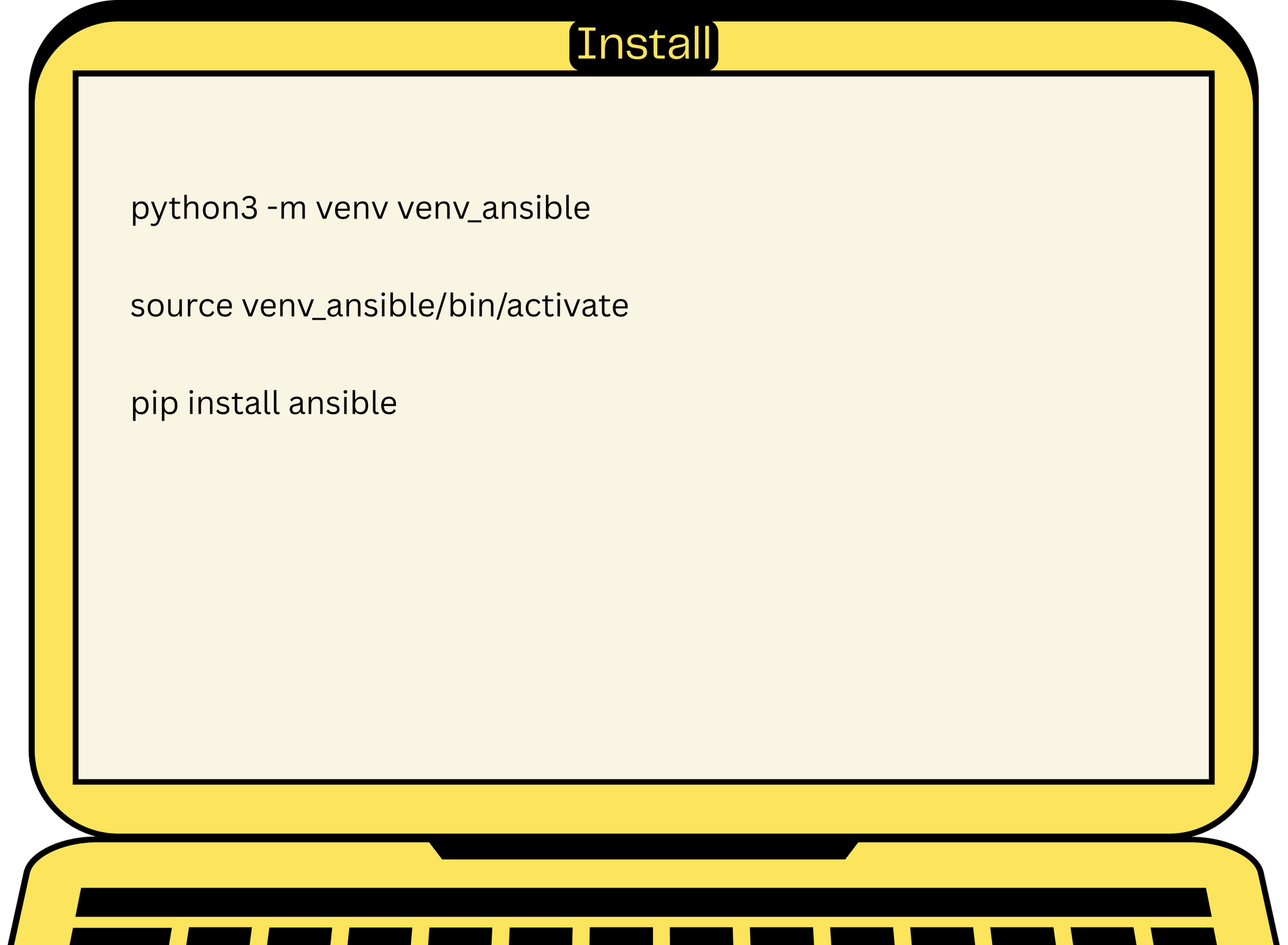
- consistentie (idempotentie)
- tijdwinst
- makkelijk op te zetten
- makkelijk te leren
- veel externe “rollen” (geerlingguy)

Hoe werkt Ansible?

- Access op basis van ssh
- De “inventory” bevat al de te managen nodes
- Vanaf een centrale "control" node configureer je diverse "managed" nodes en zorg je ervoor dat de staat van de nodes is zoals beschreven in playbooks
- De playbooks zijn in YAML
- Veel gebruik van variabelen en rollen

Voorbeeld runnen van een playbook:

```
$ ansible-playbook -C -i inventory -l webservers plays/nginx.yml
```


Install
python3 -m venv venv_ansible
source venv_ansible/bin/activate
pip install ansible

Access

Gebruikelijk, access per user:

```
$ cat /etc/sudoers.d/99-ansible-users
bob     ALL=(ALL) NOPASSWD:ALL
alice   ALL=(ALL) NOPASSWD:ALL
```

En ssh keys in home dirs.

Je ziet ook wel het gebruik van een user “ansible”.

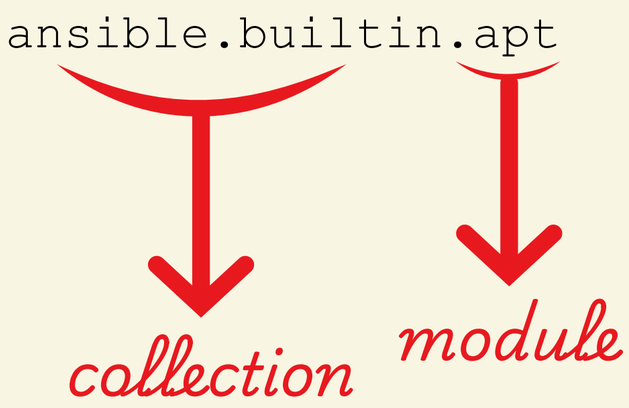
Mbt auditing beter per user

Task

“Kleinste eenheid” binnen ansible: een task

```
- name: dit is een task
  ansible.builtin.apt:
    update_cache: yes
```

Elke task gebruikt precies 1 module:



Play: hosts + Task

Een ***play*** koppelt 1 of meerdere ***tasks*** aan ***hosts***:

```
- name: dit is een play
  hosts: webservers
  become: true
  tasks:
    - name: update cache
      ansible.builtin.apt:
        update_cache: yes

    - name: install nginx
      ansible.builtin.apt:
        name: nginx
        state: present
```

Roles

De tasks kun je ook verzamelen in een role:

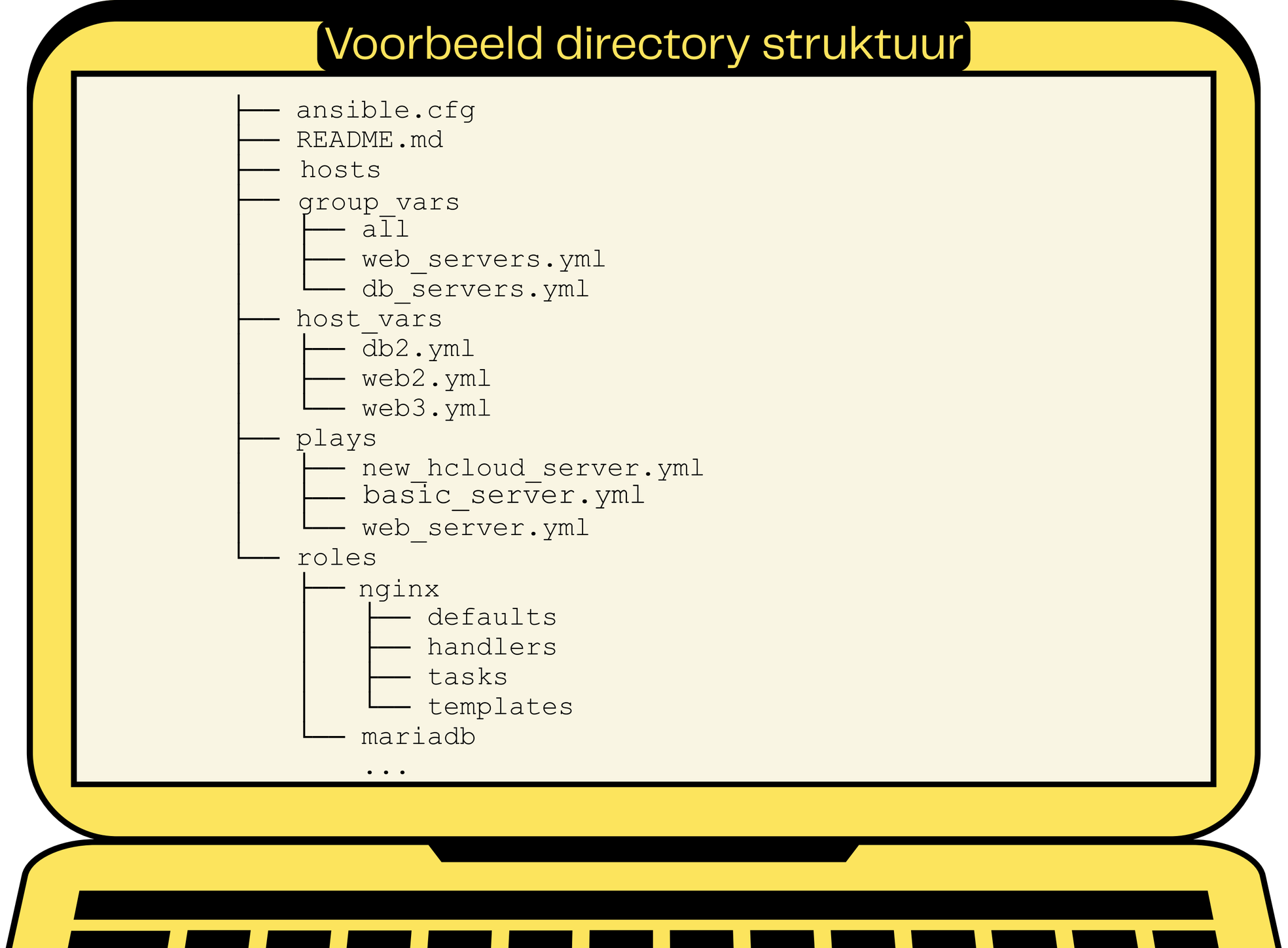
```
- name: dit is een play met een role
  hosts: webservers
  become: true
  roles:
    - name: make_ansible_ready
```

En in roles/make_ansible_ready/**tasks**/main.yml:

```
- name: Update repo cache
  ansible.builtin.apt:
    update_cache: True

- name: nog een task..
  ...
```

Voorbeeld directory struktuur



```
├── ansible.cfg
├── README.md
├── hosts
├── group_vars
│   ├── all
│   ├── web_servers.yml
│   └── db_servers.yml
├── host_vars
│   ├── db2.yml
│   ├── web2.yml
│   └── web3.yml
├── plays
│   ├── new_hcloud_server.yml
│   ├── basic_server.yml
│   └── web_server.yml
└── roles
    ├── nginx
    │   ├── defaults
    │   ├── handlers
    │   ├── tasks
    │   └── templates
    └── mariadb
        ...
```

ansible.cfg

Terug

```
$ cat ansible.cfg
[defaults]
roles_path=./roles/external:.roles/internal
log_path = /var/log/ansible.log
vault_password_file = .vault_pass
ansible_managed = "Ansible managed: no local edits!"
connect_timeout = 120
nocows = True

[privilege_escalation]
become = True
become_method = sudo
```


inventory
Terug
$ cat hosts
[all:children]
production
develop
[production]
web1.acme.com
web2.acme.com
db1.acme.com
[develop]
web_dev.acme.com
db_dev.acme.com

group_vars en host_vars

Terug

```
group_vars
├── all
│   ├── vars
│   │   └── main.yml
│   └── vault
│       └── main.yml
└── webservers
    └── vars
        └── main.yml
host_vars
└── ai.solarcontrol.nl
    └── vars
        └── main.yml
```

plays/basic_server.yml

Terug

```
---
- name: Install and config a basic server
  hosts: all
  gather_facts: True
  become: yes
  roles:
    - name: make_ansible_ready
    - name: ufw
    - name: standard_packages
    - name: hardening
    - name: monitoring
```

role nginx

Terug

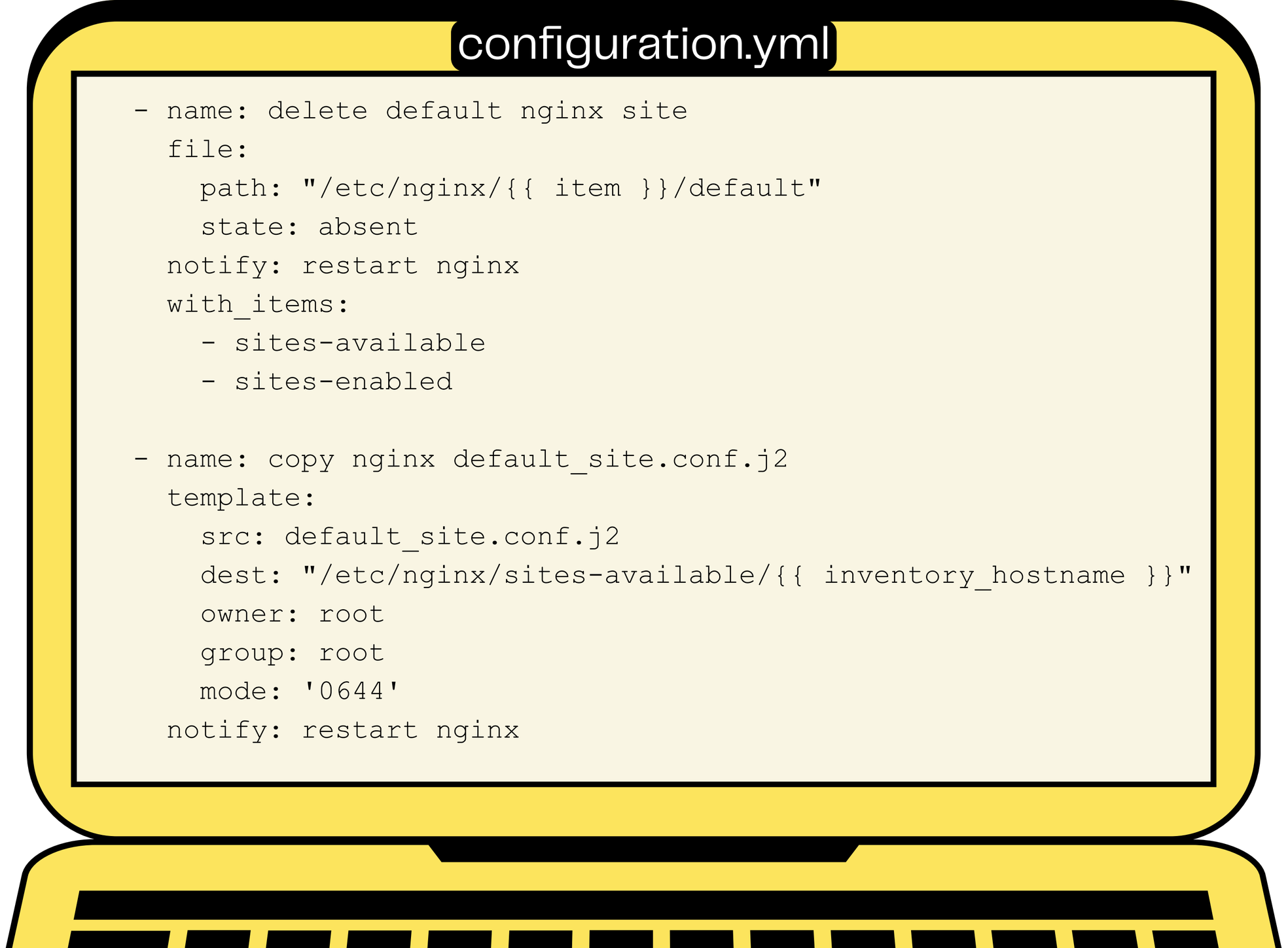
```
nginx/
├── templates
│   ├── index.html.j2
│   └── default_site.conf.j2
├── tasks
│   ├── main.yml
│   ├── installation.yml
│   └── configuration.yml
├── handlers
│   └── main.yml
└── defaults
    └── main.yml
```

installation.yml

```
---
- name: Update repo cache
  apt:
    update_cache: yes
- name: Install packages
  apt:
    name: "{{ nginx_packages }}"
    state: present
```

Variabele ”nginx_packages” in roles/nginx/default main.yml:

```
---
nginx_packages:
  - nginx
nginx_user: www-data
nginx_group: www-data
```


configuration.yml
- name: delete default nginx site
file:
path: "/etc/nginx/{{ item }}/default"
state: absent
notify: restart nginx
with_items:
- sites-available
- sites-enabled
- name: copy nginx default_site.conf.j2
template:
src: default_site.conf.j2
dest: "/etc/nginx/sites-available/{{ inventory_hostname }}"
owner: root
group: root
mode: '0644'
notify: restart nginx

Variabelen

Variabelen kom je op veel plekken tegen.
In volgorde van voorkeur (van laag naar hoog)

- role defaults
- group_vars/ (all, group)
- inventory vars
- host_vars/ (all, host)
- host facts
- task vars
- set_facts
- extra vars (via command line: ”-e var=value”

Zie docs.ansible.com voor een meer uitgebreid overzicht van de volgorde van variabelen

Variabele in een task

```
- name: variabele in een task
  hosts: ai.solarcontrol.nl
  gather_facts: no
  tasks:
    - name: use vars
      debug:
        msg: "{{ my_var }}"
      vars:
        my_var: "hallo"
```

set_fact

Toon alle host ”facts” met:

```
ansible -i hosts -m setup ai.solarcontrol.nl
```

```
- name: facts
  hosts: ai.solarcontrol.nl
  gather_facts: yes
  tasks:
    - name: Creating a dictionary var
      ansible.builtin.set_fact:
        event_dict: {'event': linuxnijmegen, 'datum': '2025-05-12'}

    - name: reboot at "{{ event_dict.datum }}"
      ansible.builtin.reboot:
        msg: "Rebooting {{ inventory_hostname }}"
      when: ansible_date_time.date == event_dict.datum
```

Vragen?

Terug

Bier?