What is Ansible?

Ansible is an open-source tool that helps you automate things like:

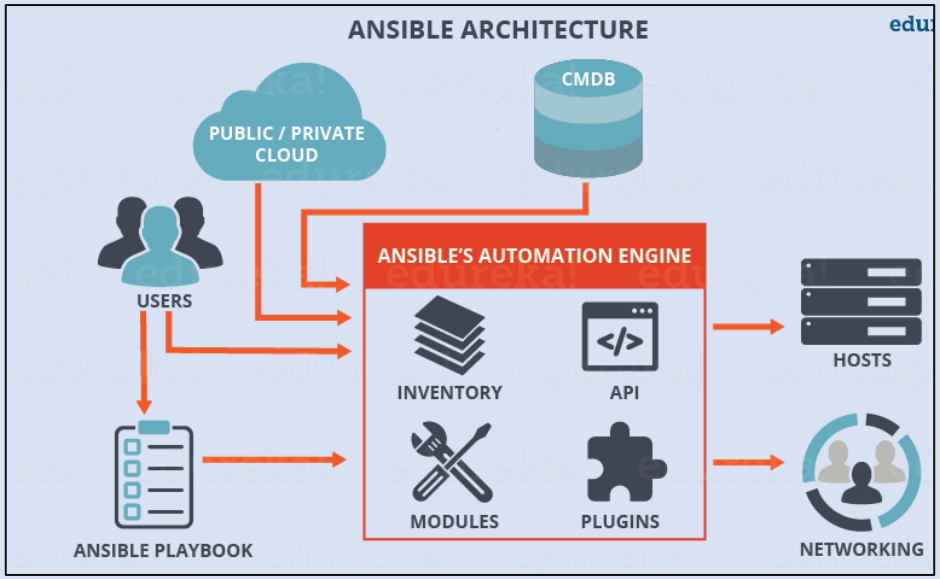
- Installing software
- Updating systems
- Managing many computers at once

You write simple instructions in a file (called a **playbook**) using easy-to-read language (**YAML**), and ansible does the rest.

Think of it like:

Instead of doing the same task on 10 servers one by one, Ansible lets you do it all at once with just one command!

Pawar Divya



1. **Users**
    - These are the people (like you) who write **Ansible Playbooks** to define what tasks should be automated.
2. **Ansible Playbook**
    - A file written in **YAML** that contains step-by-step instructions for what Ansible should do (e.g., install software, start a service).

Pawar Divya

3. **CMDB & Cloud**
    - CMDB (Configuration Management Database) and cloud platforms (like AWS, Azure, etc.) provide **information about systems** (like their IP addresses, OS, etc.) to Ansible.
4. **Ansible Automation Engine**

    This is the core of Ansible where all the action happens. It has several components:
    - **Inventory:** List of server or systems Ansible will manage (called "hosts").
    - **Modules:** Small programs that do specific tasks like installing packages or copying files.
    - **Plugins:** Add extra features like logging or connecting to cloud platforms.
    - **API:** Allows Ansible to connect with other tools or systems.

Pawar Divya

5. **Output (Target Systems)**

   Ansible performs actions on:

   - **Hosts (Servers):** Like installing software, updating packages.
   - **Networking devices:** like routers and switches.

Creating an Ansible Playbook for Apache Installation

As for the master-worker setup (referring to a control node and managed nodes in an Ansible setup):

Steps for Master-Worker Setup

1. **Install Ansible on the Control plane (Master)**
   - Update system: sudo apt update
   - Install Ansible: sudo apt install ansible

2. Configure SSH Access

- Ensure passwordless SSH access from the control node (master) to the worker node (managed) node.
- Generate SSH key pair on the control plane: ssh-keygen
- Copy the SSH public key to the worker node: ssh-copy-id user@worker-node

```
controlplane:~$ ssh-copy-id root@172.30.2.2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_ed25519.pub"
The authenticity of host '172.30.2.2 (172.30.2.2)' can't be established.
ED25519 key fingerprint is SHA256:XikjTSvie4xfSHl8CD14UKeq6b3m5zuJAlVVC6JLlao.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:2: [hashed name]
    ~/.ssh/known_hosts:3: [hashed name]
    ~/.ssh/known_hosts:4: [hashed name]
    ~/.ssh/known_hosts:5: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any tha
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it i

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@172.30.2.2'"
and check to make sure that only the key(s) you wanted were added.

controlplane:~$
```

Pawar Divya

3. Create Ansible Inventory File

```
controlplane:~$ sudo mkdir /etc/ansible -p
controlplane:~$ sudo vi /etc/ansible/hosts
controlplane:~$
```

Add this:

```
[workers]
root ansible_host=172.30.2.2 ansible_user=root
```

4. Test SSH Connection via Ansible

```
controlplane:~$ ansible workers -m ping
root | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
controlplane:~$
```

Pawar Divya

5. Create Apache Installation Playbook:

```yaml
---
- name: Install and Start Apache on Ubuntu
  hosts: workers
  become: yes

  tasks:
    - name: Update apt cache
      apt:
        update_cache: yes

    - name: Install Apache
      apt:
        name: apache2
        state: present

    - name: Start and enable Apache service
      service:
        name: apache2
        state: started
        enabled: yes

    - name: Create a simple index.html
      copy:
        dest: /var/www/html/index.html
        content: "Hello from Control Plane ANSIBLE!"
```

6. Run the Playbook

```
controlplane:~$ ansible-playbook apache-install.yml

PLAY [Install and Start Apache on Ubuntu] ***************************************

TASK [Gathering Facts] **********************************************************
ok: [node01]

TASK [Update apt cache] *********************************************************
changed: [node01]

TASK [Install Apache] ***********************************************************
ok: [node01]

TASK [Start and enable Apache service] ******************************************
ok: [node01]

TASK [Create a simple index.html] ***********************************************
changed: [node01]

PLAY RECAP **********************************************************************
node01                     : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

controlplane:~$
```

Pawar Divya

7. Verify on Worker Node

```
controlplane:~$ ssh node01
Last login: Thu Apr 24 12:16:39 2025 from 10.244.5.198
node01:~$ curl localhost
Hello from Control Plane ANSIBLE!node01:~$
```

In this task, we installed Apache on the control node, created a playbook, and used it to install and configure Apache on the worker node with a custom message. This demonstrates how Ansible makes it easy to automate tasks across multiple servers.

Thank you!

Pawar Divya