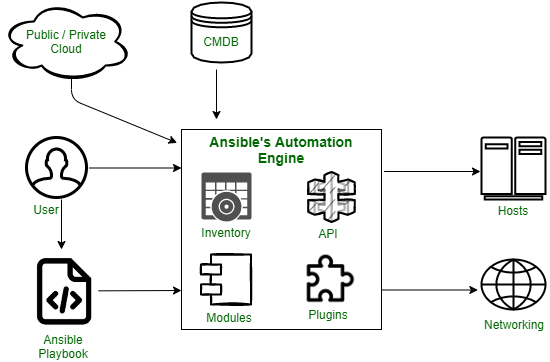
**Ansible and its Architecture components**

* Ansible is an **open-source automation tool** used to manage and configure servers, deploy applications, and automate repetitive IT tasks.
* Ansible is **agentless**, meaning it does not require software to be installed on managed nodes it uses SSH to communicate.
* Tasks and configurations are defined in simple, human-readable YAML files called **playbooks**.
* It can manage infrastructure of any size and integrates with cloud providers, containers, and DevOps tools.
* Originally developed by **Michael DeHaan** in 2012, Ansible was later acquired by **Red Hat** in 2015. It is widely used across **Linux, Unix-like systems, macOS, and Windows**

**Architecture components:**



* **Inventories -**Ansible inventories are lists of hosts with their IP addresses, servers, and databases which have to be managed via an SSH for UNIX, Linux, or Networking devices, and WinRM for Windows systems.
* **APIs -**Application Programming Interface or APIs are used as a mode of transport for public and private cloud services.
* **Modules -**Ansible modules are executed on remote hosts via playbooks to manage resources like services, packages, files, and system commands. With over 450 modules, they automate various tasks, including cloud operations—for example, Cloud Modules like CloudFormation can create or delete AWS stacks.
* **Plugins -**Extend Ansible’s core functions; they can execute tasks and act as interfaces to modules. Custom plugins can also be created.
* **Networking -**Ansible uses a simple, powerful, and agent-less automation framework to automate network tasks.
* **Hosts -**Hosts refer to the nodes or systems (Linux, Windows, etc) which are automated by Ansible.
* **Playbooks -**Playbooks are simple files written in YAML format which describe the tasks to be executed by Ansible.
* **CMDB A Configuration Management Database that stores and manages data about IT assets and their relationships.**
* **Cloud -**A remote network of servers used for storing, managing, and processing data online instead of locally.

**Advantages:**

1. No need to install anything on client machines.
2. Easy to write and understand using simple YAML files.
3. Safe to run multiple times without breaking anything.
4. Works on many systems like Linux, Windows, and cloud servers.
5. Code is reusable and fits well in big projects.
6. Works with tools like Jenkins, Docker, and Git.
7. Keeps systems secure using SSH and encryption.

**Disadvantages:**

1. Has limited or paid graphical interface (GUI).
2. Can be slow with large number of systems.
3. Works better on Linux than on Windows.
4. Error messages can be hard to understand.
5. Doesn’t undo changes automatically if something fails.

**Note: This notes contains only the definition and components of Ansible, To write full Architecture write with Ansible Master– Slave Configuration.**