DevOps

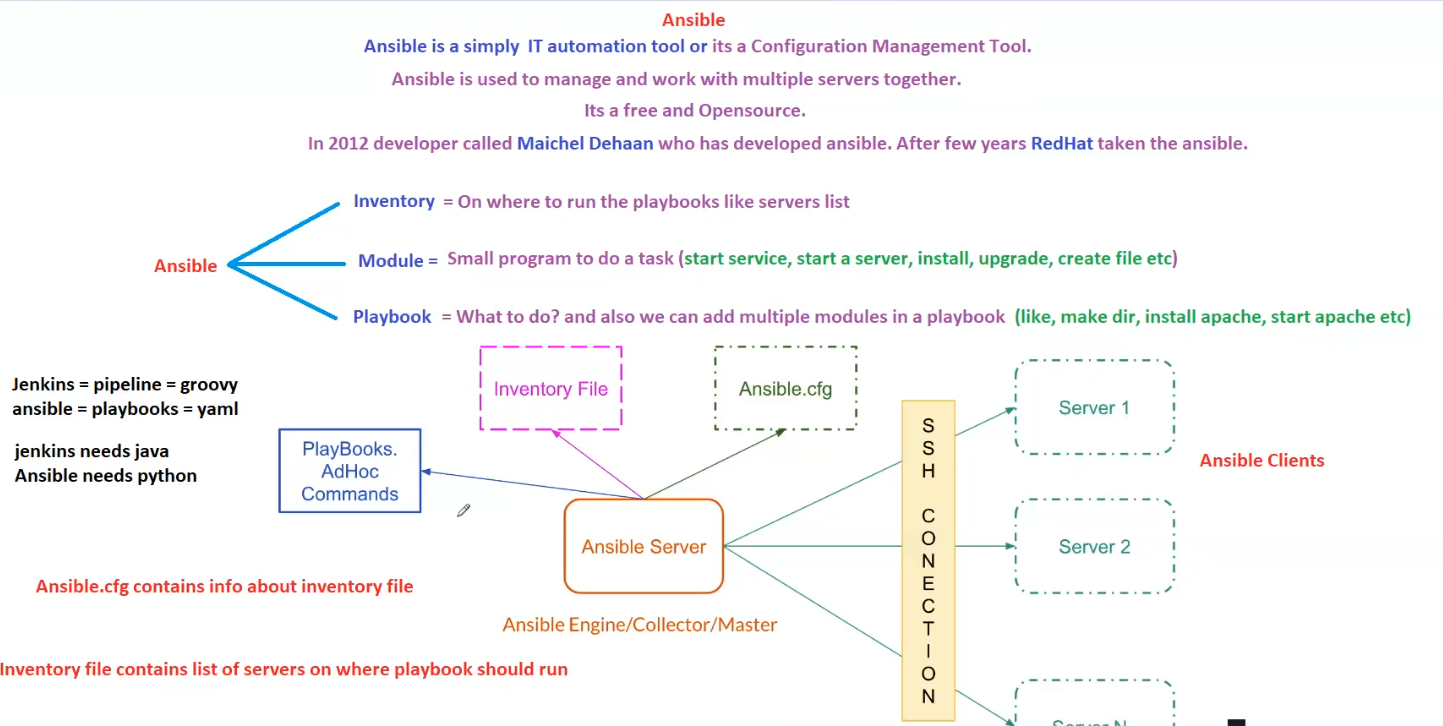
[Ansible 2](#_Toc194360029)

[Ansible Overview 2](#_Toc194360030)

[Ansible: Manual vs. Automation 4](#_Toc194360031)

# Ansible

## Ansible Overview



Ansible is a simple **IT automation tool** and a **Configuration Management Tool**. It is used to manage and work with multiple servers simultaneously.

**Key Features:**

* **Free and Open-source**
* Developed in **2012** by **Maichel Dehaan**
* Later acquired by **Red Hat**

**Core Components of Ansible**

**1. Inventory**

* Defines where to run the playbooks (a list of servers).
* The **Inventory file** contains the list of servers where the playbook should execute.

**2. Module**

* A **small program** that performs specific tasks such as:
  + **Start a service**
  + **Start a server**
  + **Install software**
  + **Upgrade software**
  + **Create files**

**3. Playbook**

* Defines **what to do** in Ansible automation.
* Written in **YAML** format.
* Can include multiple **modules** to perform various tasks such as:
  + **Create directories**
  + **Install Apache**
  + **Start Apache service**

**Ansible Architecture**

**1. Ansible Server (Master Node)**

* Acts as the **Ansible Engine/Collector/Master**.
* Executes playbooks and commands on client machines.
* Requires **Python** to run.

**2. Ansible Clients (Managed Nodes)**

* Systems managed by the Ansible server.
* Connected via **SSH** for remote execution.

**3. Configuration Files**

**a) ansible.cfg**

* Contains configuration details, including information about the **Inventory file**.

**b) Inventory File**

* Lists servers where playbooks should run.

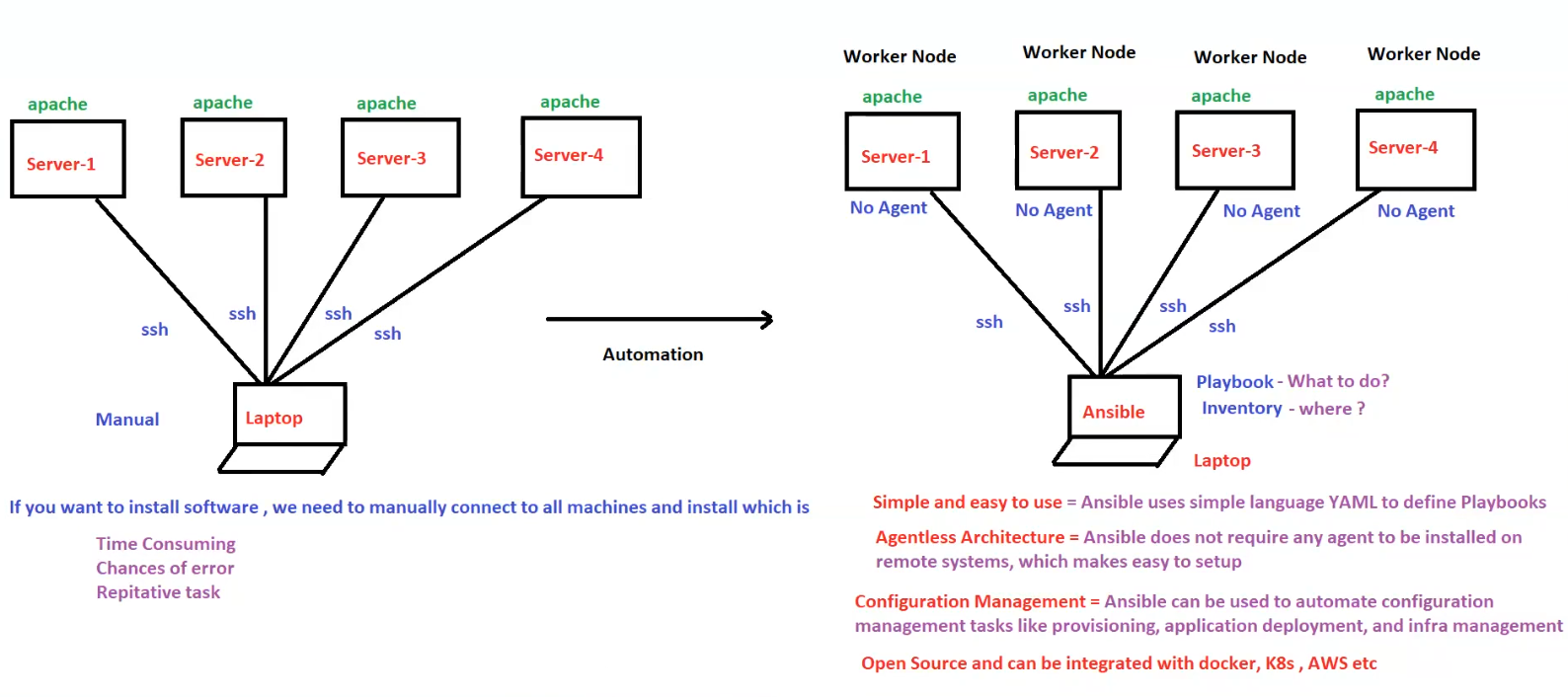
**Execution Flow in Ansible**

1. The **Ansible Server** loads the **Inventory File** and **ansible.cfg**.
2. It uses **Playbooks or AdHoc Commands** to define the tasks.
3. Establishes an **SSH connection** to the **Ansible Clients** (servers).
4. Executes the defined tasks using **Ansible Modules**.

**Comparison with Jenkins**

| **Feature** | **Jenkins** | **Ansible** |
| --- | --- | --- |
| **Used For** | CI/CD Pipelines | Configuration Management & Automation |
| **Scripting Language** | Groovy (Pipeline) | YAML (Playbooks) |
| **Dependency** | Requires Java | Requires Python |

## Ansible: Manual vs. Automation



**1. Manual Software Installation (Before Automation)**

* To install software (e.g., Apache) on multiple servers, we must **manually** connect to each machine via **SSH** and execute commands.
* This approach has several drawbacks:
  + **Time-consuming**
  + **Prone to errors**
  + **Repetitive task**

**Manual Setup**

* A laptop (control machine) connects to multiple servers manually using SSH.
* Software installation and configuration are done individually on each server.

**2. Automated Configuration Management using Ansible**

**How Ansible Automates the Process:**

* The **Ansible control node** (Laptop) manages multiple servers (**Worker Nodes**) using **Playbooks** and **Inventory files**.
* Connections are established over **SSH**, and there is **no need to install an agent** on worker nodes.

**Key Benefits of Ansible Automation**

1. **Simple and Easy to Use**
   * Uses **YAML** to define playbooks.
   * Playbooks describe **what to do** (tasks) and **where to do it** (inventory).
2. **Agentless Architecture**
   * Unlike other automation tools, Ansible **does not require an agent** on remote systems.
   * Simplifies setup and reduces overhead.
3. **Configuration Management**
   * Automates provisioning, application deployment, and infrastructure management.
4. **Open Source & Integration**
   * Ansible is open-source and supports integrations with **Docker, Kubernetes (K8s), AWS**, and more.

**Comparison: Manual vs. Ansible Automation**

| **Feature** | **Manual Process** | **Ansible Automation** |
| --- | --- | --- |
| **Execution Method** | Manually logging into each server | Uses Playbooks & SSH |
| **Time Consumption** | High | Low |
| **Risk of Errors** | High | Low |
| **Scalability** | Difficult to scale | Easily scalable |
| **Agent Requirement** | Not required | Not required (Agentless) |
| **Configuration Management** | No automation | Fully automated |

**Conclusion**

Ansible simplifies server configuration and application deployment by automating repetitive tasks. Its **agentless architecture**, **YAML-based playbooks**, and **integration with modern technologies** make it a powerful tool for **DevOps** and **IT automation**. 🚀