Configuration Management

* Configuration Management (CM) is the process of automating, tracking, and maintaining system configurations to ensure consistency, stability, and efficiency across IT infrastructure.
* ***Ensures consistency*** across multiple servers: In a large IT infrastructure, multiple servers need to have the same configuration to ensure smooth operation. Configuration Management (CM) tools automate
* ***Automates system setup*** (software installation, security patches, configurations).
* ***Prevents configuration drift***: Configuration drift occurs when identical servers gradually become inconsistent due to manual interventions, inconsistent updates, or missed patches. This results in varying performance and potential security risks,

Configuration Management Tools

**1️⃣ Ansible (Agentless)**

**2️⃣ Puppet (Agent-Based)**

**3️⃣ Chef (Agent-Based)**

**4️⃣ SaltStack (Hybrid)**

**Agentless Architecture** refers to a system or infrastructure where no software agent (a small program that runs in the background on a machine or system) is required to be installed or run on the target systems. Instead of agents, the system uses other methods, such as APIs, SSH, or other network protocols, to interact with and manage the systems.

 an **agent** refers to a piece of software or a process that is installed on a target machine (also called a managed or remote host) to allow communication between the automation tool and the target machine.

1. **Ansible** (Agentless)

* Open-source automation tool developed by Red Hat
* Agentless (No need to install software on managed nodes)
* Communicates using SSH (Linux) and WinRM (Windows)

1. **Puppet (Agent-Based)**

* Developed by Puppet Labs
* Uses Declarative DSL (Domain Specific Language)
* Uses a Client-Server Model.

1. **Chef (Agent-Based)**

* Developed by Progress Chef
* Uses Ruby DSL for writing configuration scripts
* Requires an Agent

1. **SaltStack (Hybrid)**

* Developed by SaltStack (Acquired by VMware)
* Supports both agent-based and agentless modes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature** | **Ansible** | **Puppet** | **Chef** | **SaltStack** |
| **Architecture** | Agentless | Agent-based | Agent-based | Hybrid |
| **Language** | YAML (Playbooks) YAML Ain't Markup Language | Puppet DSL | Ruby DSL | YAML + Jinja |
| **Setup Complexity** | Easy | Medium | Complex | Medium |
| **Execution Speed** | Slow | Medium | Medium | Fast |
| **Scalability** | Medium | High | High | Very High |
| **Ease of Learning** | Easy | Moderate | Hard | Moderate |

**Which Configuration Management (CM) Tool Is Most Widely Used and Why?**

* **Ansible** is currently the most widely used **Configuration Management (CM) tool** in the industry.
* Unlike Puppet and Chef, which require an **agent to be installed on each managed system**, Ansible **does not need an agent**.
* It uses **SSH for Linux servers** and **WinRM for Windows**, making deployment **simpler and more lightweight**.
* **Ansible uses YAML (Playbooks),** which is **easy to read and write**, compared to Puppet’s DSL or Chef’s Ruby DSL.
* **No programming knowledge is required**
* **Ansible is easier to install and set up** compared to Puppet and Chef, which require complex master-agent configurations.

Installation of Ansible

For practising ansible two servers should created

1. Ansible server/Control node : which you execute your Ansible commands and manage configurations across your infrastructure. It is where Ansible is installed and where you run your playbooks, modules, and other commands
2. Target server:  refers to the system or machine that is being configured or managed by Ansible, These nodes receive and execute tasks from the control node.

***Install Ansible***

* sudo apt update

Sudo: It allows a permitted user to **run commands as a superuser (root)**

* sudo apt install ansible
* Add Ansible PPA (Personal Package Archive): This step ensures that you get the latest stable version of Ansible.

 sudo add-apt-repository --yes --update ppa:ansible/ansible

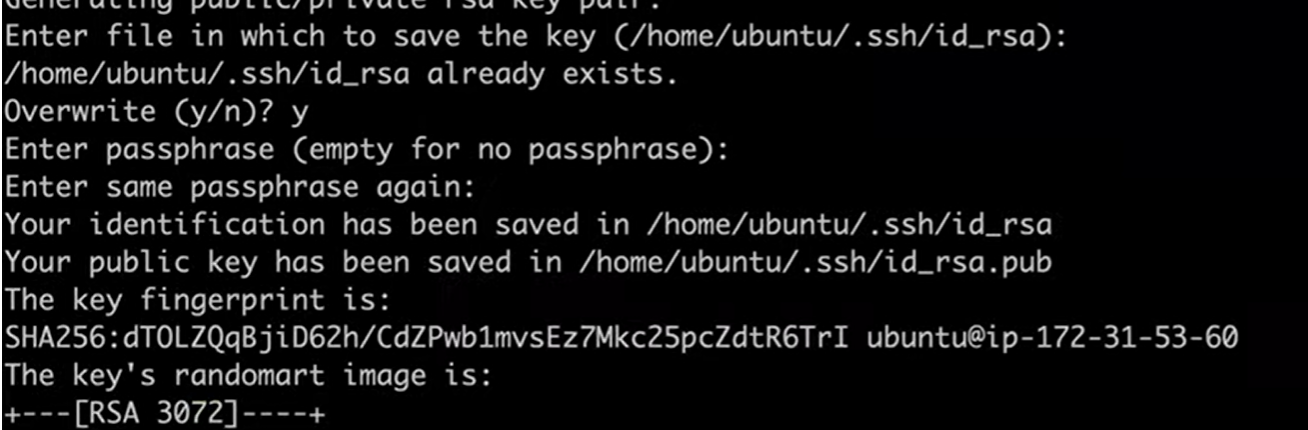
* ansible --version

Once installation done set a passwordless authentication.

Passwordless authentication is a method of logging into a system or server without the need to manually enter a password

* Login to the main server
* Ssh-keygen

This will create an public key aswell private key, then just give enter it will ask to path to download the key

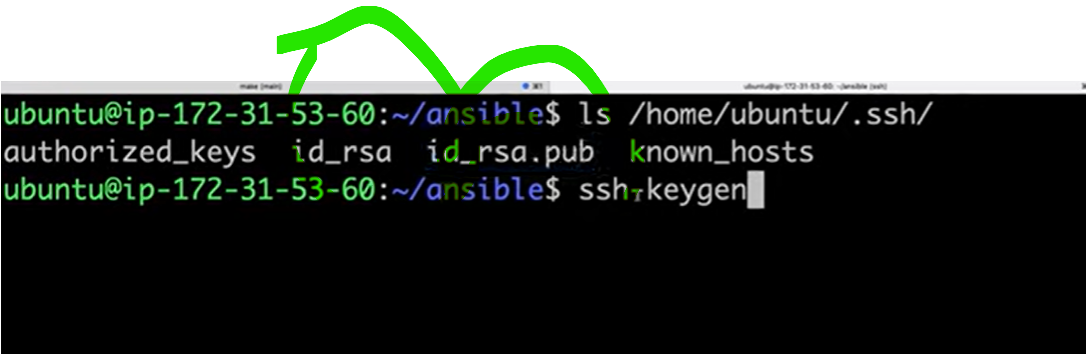


* Login in to the directory where the keys download

             Pub is the public key

* Copy that public key content

  Give cat filename.pub



* Now lgin to the target server
* Go to the ssh folder → cd/home/ubuntu/.ssh
* Paste the Public key which we copied from the client server in authorized keys
* One more way to copy this public key to authorised through cmd  
  ssh-copy-id -i /home/ubuntu/.ssh/id\_ed25519.pub ubuntu@172.31.81.84
* Then go back to the main server give ssh ip address(either private or public)