**Explain the master slave configuration of ansible:**

**Ansible Installation:** In order to install ansible, the system must have python preinstalled.**Open up Terminal and execute the following commands:**

**1. Add the Ansible Repository.**

sudo apt-add-repository -y ppa:ansible/ansible

**2. Update the system repository listings.**

sudo apt-get update

**3. Install the ansible packages.**

sudo apt-get install -y ansible

**Ansible Master-Slave Configuration:**

**1. What is the Master?**

The Master, or Control Node, is the central machine where Ansible is installed and executed.

**Key Points:**

* It is responsible for writing playbooks, defining inventories, and sending instructions to slave nodes.
* It can be any Linux/Unix system (Ubuntu, CentOS, etc.) that has Python installed.
* Contains important Ansible files like ansible.cfg, inventory (hosts), and all playbooks/scripts.
* A single master can manage 10s, 100s, or even 1000s of slave nodes efficiently.

**2. What is a Slave?**

The Slave, also called the Managed Node, is any system that the master controls using SSH.

**Key Points:**

* The slave does not require Ansible or any agent to be installed — just SSH and Python.
* The tasks written in playbooks are executed on these nodes via SSH.
* Any Platform like Linux, Windows (with WinRM), virtual machines, Docker containers, or even cloud instances (AWS EC2, Azure VMs).
* You can easily add or remove managed nodes just by updating the inventory file.

**3. How Master Connects to Slaves (SSH Connection):**

Ansible uses SSH protocol to securely connect to slaves and run commands remotely.

**Key Points:**

* Use ssh-keygen and ssh-copy-id to set up key-based authentication.
* SSH uses port 22; you can customize ports for each node in the inventory file.
* The master controls all connections, so there’s no need for inbound traffic on the slaves.
* No need for agents or background daemons like in other tools (e.g., Puppet, Chef).

**4. Inventory File (Defining Slaves):**

The inventory file (usually named hosts) is a plain-text file that lists all the IP addresses or domain names of managed nodes.

**Key Points:**

* Servers can be grouped logically (e.g., [web], [db], [all\_servers]) for better control.
* You can define variables specific to hosts or groups (like ports, user, etc.).
* Ansible can also pull inventory from cloud providers using dynamic inventory scripts or plugins.
* Inventories can be written in different formats depending on complexity INI, YAML, or JSON.

**Example:**

[web]

192.168.1.10

192.168.1.11

**[db]**

192.168.1.20 ansible\_user=postgres ansible\_port=2222

**5. Playbooks (Ansible Scripts):**

Playbooks are written in YAML format and contain the automation logic – what tasks to perform on which group of servers.

**Key Points:**

* Easy to read and write due to simple YAML syntax.
* You define *what* you want to happen, not how to do it step-by-step.
* Playbooks can include roles, handlers, variables, and templates for clean, reusable code.
* You can install software, create users, configure firewalls, restart services, etc., all in one file.

**Example:**

**- name: Install NGINX**

**hosts: web**

**become: yes**

**tasks:**

**- name: Install NGINX package**

**apt:**

**name: nginx**

**state: present**

**6. Execution Flow (How Everything Works):**

This is the step-by-step process of how the Ansible master interacts with slaves.

**Key Points:**

* You run a command like ansible-playbook site.yml from the master.
* Ansible reads the inventory file to determine the target machines.
* It connects to all listed nodes using SSH and begins executing tasks.
* Tasks are run line by line, and output/results are displayed on the master terminal.

**7. Configuration Files (Supporting Files in Master):**

Along with the playbook and inventory, the Ansible master uses a few important configuration files.

**Key Points:**

* ansible.cfg: Central config file to define SSH settings, roles path, retries, and other behavior.
* hosts: The inventory file that lists all target nodes.
* Custom Modules: You can write your own modules and store them in the designated path.
* Templates & Handlers: Used to manage dynamic configuration files and event-triggered actions.

**Advantages of Ansible Master-Slave Configuration:**

1. No agents needed on slave machines – only SSH is required.
2. Easy to set up and use with simple YAML playbooks.
3. Centralized control of multiple machines from one master.
4. Highly scalable to manage hundreds or thousands of servers.
5. Supports various platforms including cloud, Linux, and Windows.

**Disadvantages of Ansible Master-Slave Configuration:**

1. Requires Python on all managed nodes by default.
2. Performance may slow down with very large inventories.
3. Limited Windows support compared to Linux.
4. No built-in scheduling like some other automation tools.
5. Debugging complex playbooks can be time-consuming.