

Introduction to Ansible - Basic Study Material

What is Ansible?

Ansible is an open-source automation tool used for configuration management, application deployment, and task automation. It allows users to automate IT infrastructure with simple, human-readable scripts written in YAML.

Core Components of Ansible

1. Inventory

The inventory file is a list of managed hosts stored in a file. This file can be in **INI** or **YAML** format. By default, Ansible uses the inventory located at `/etc/ansible/hosts`.

Example of Inventory File (INI format)

```
[webservers]
web1.example.com
web2.example.com
```

```
[dbservers]
db1.example.com
db2.example.com
```

In the above example, there are two groups: `webservers` and `dbservers`, each containing multiple hosts.

2. Modules

Modules are scripts that perform specific tasks like installing packages, copying files, or restarting services. Each module accepts parameters and returns output in JSON format.

Example of a Module Usage

The following command installs `nginx` on a remote machine:

```
ansible all -m apt -a "name=nginx state=present" -b
```

3. Variables

Variables help in managing system differences and making playbooks dynamic. Variables can be stored in **dictionaries** or **lists**.

Example of Variables in Playbooks

```
vars:
  database_name: mydb
  destination: /etc/config/
```

Variables can also be grouped based on host or playbook level.

4. Facts

Facts are system information automatically gathered by Ansible. These include OS type, IP addresses, and memory usage.

Example of Fact Gathering

To display facts about a host, run:

```
ansible all -m setup
```

5. Playbooks

Playbooks define automation tasks in **YAML format**. They contain multiple plays that map groups of hosts to specific tasks.

Example of a Simple Playbook

```
- name: Install and start Apache
```

```
  hosts: webservers
```

```
  become: yes
```

```
  tasks:
```

```
    - name: Install Apache
```

```
      apt:
```

```
        name: apache2
```

```
        state: present
```

```
    - name: Start Apache
```

```
      service:
```

```
        name: apache2
```

```
        state: started
```

6. Configuration File

Ansible uses a configuration file (ansible.cfg) to override default settings. The order of configuration file lookup is:

1. ansible.cfg in the current directory
2. ~/.ansible.cfg in the home directory
3. /etc/ansible/ansible.cfg (default)

Example of an Ansible Configuration File

```
[defaults]
```

```
inventory = ./inventory
```

```
host_key_checking = False
```

```
retry_files_enabled = False
```

Summary Table

Component	Description
Inventory	List of managed hosts (INI/YAML format)
Modules	Predefined tasks executed by Ansible
Variables	Store data dynamically for tasks
Facts	Automatically gathered system information
Playbooks	YAML-based automation scripts

Configuration File Controls Ansible's default behavior

Conclusion

Ansible simplifies automation using YAML-based playbooks and eliminates the need for manual configuration. Understanding these basic components will help in efficiently managing and automating IT tasks.