Ansible: Zero to Hero - Notes

What is Ansible?

Ansible is a tool used for configuration management, automation, and orchestration.

It allows you to automate tasks such as installing packages, managing files, and starting/stopping services on multiple servers.

Prerequisites

- Minimum 2 servers needed:

  1. Main Server (Control Node - where Ansible is installed)

  2. Target Server(s) (Managed Nodes)

- Install Ansible on main server:

```
  sudo apt install ansible
```

- Password-less SSH authentication is required:

```
  ssh-keygen
  ssh-copy-id user@target_server
```

Inventory File

An inventory file contains the IPs or hostnames of the target servers.

Example:

```
[webservers]
192.168.1.2
```

```
192.168.1.3


[dbservers]

192.168.1.4

192.168.1.5
```

Running Shell Commands with Ansible

Run ad-hoc commands without writing a playbook:

Example:

```
ansible -i inventory all -m "shell" -a "uptime"
```

To check disk usage on a group:

```
ansible -i inventory webservers -m "shell" -a "df -h"
```

Ansible Playbooks

- A YAML file that contains one or more tasks to run on the target systems.
- Equivalent to shell scripts in Linux, but for Ansible.

Sample Playbook: Install and Start NGINX

```
---

- name: Install and start Nginx

  hosts: all
  become: true

  tasks:
    - name: Install nginx
      apt:
        name: nginx
        state: present

    - name: Start nginx
      service:
        name: nginx
        state: started

# Alternative using shell:
    - name: Install nginx
      shell: apt install nginx -y
```

Run Ansible Playbook

```
ansible-playbook -i inventory first-playbook.yml
```

Write Files on Target Servers

Command:

```
ansible -i inventory 172.31.58.21 -m "shell" -a "touch devopsclass"
```

Ansible: Zero to Hero - Notes

Ad-hoc Commands vs Playbooks

Ad-hoc Commands:

- Used for quick one or two commands
- CLI based
- Not reusable

Playbooks:

- Used for complex or multiple tasks
- Written in YAML
- Reusable