Deploying website in Ansible (Master and Slave)

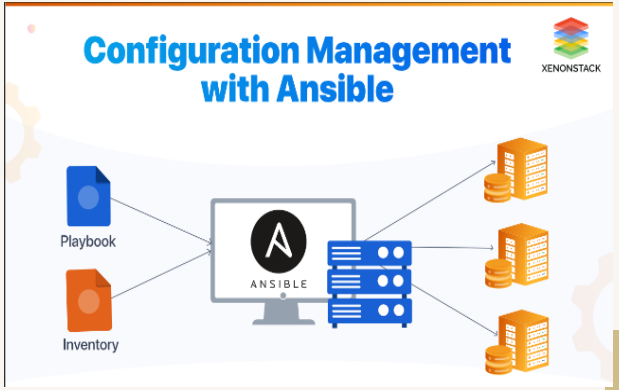
BY
KORADA VIJAYA ANJALI
ADITYA UNIVERSITY
21A91A6148

INDEX

01. Introduction

Ansible: Ansible is an open-source automation tool that is used for configuration management, application deployment, task automation, and IT orchestration. It simplifies the management of servers, applications, and infrastructure by allowing users to define and automate tasks in the form of playbooks.

- Ansible is commonly used for server provisioning, application deployment, configuration management, and various IT automation tasks. It is platform-agnostic and supports a wide range of operating systems, making it a popular choice for DevOps and system administrators to streamline and automate their tasks.



02.TECHNOLOGIES USED

1. AWS
2. EC2
3. PYTHON SERVER
4. ANSIBLE
5. INVENTORY FILE
6. PLAY BOOK
7. VS CODE

03. ANSIBLE

Control Machine or Master: Ansible control nodes are primarily used to run tasks on managed hosts. You can use any machine with Python installed as an Ansible control node.

Slave or Nodes: Ansible typically refers to machines or systems as "hosts" or "managed nodes." These are the machines that Ansible connects to and performs tasks on.

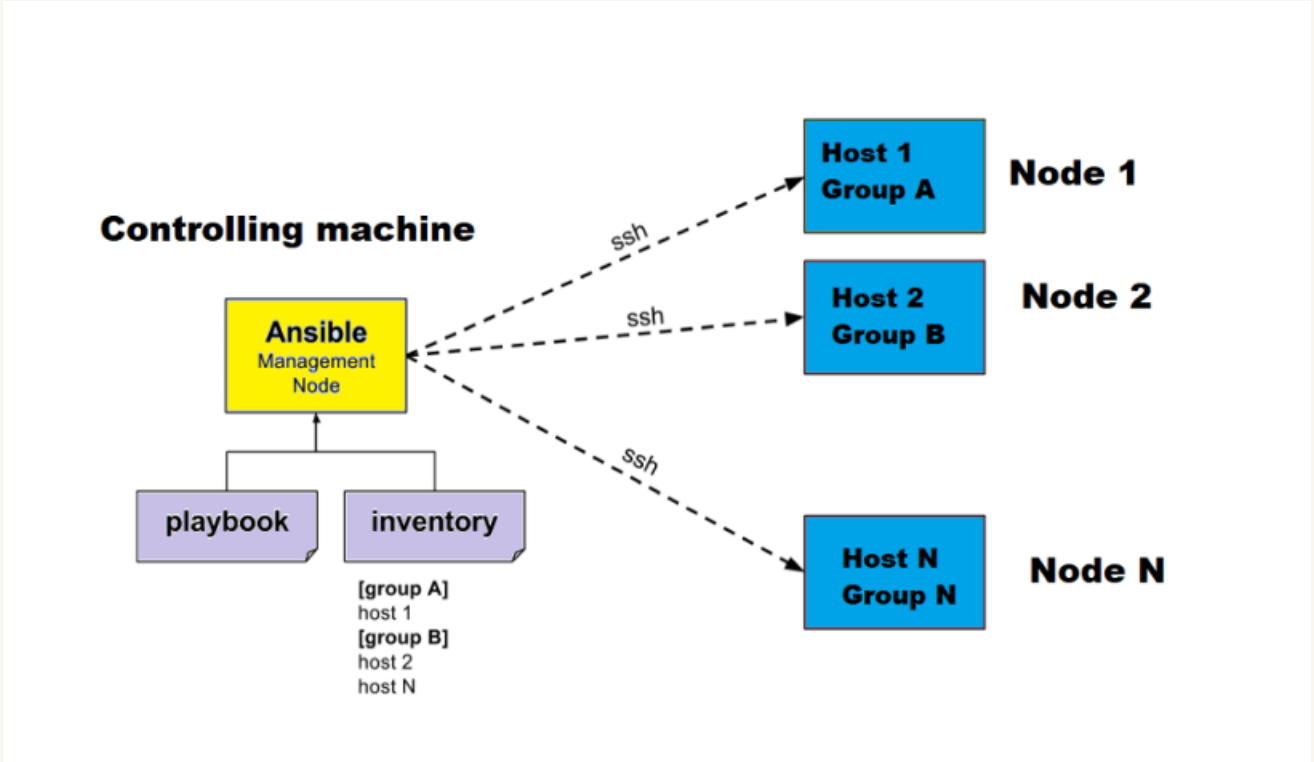
Playbooks: Playbooks are files written in YAML (Yet Another Markup Language) that define a set of tasks to be executed on target hosts.

Inventory: An Ansible inventory is a list of hosts (servers) that Ansible can manage. It helps Ansible identify which hosts to target with specific tasks.

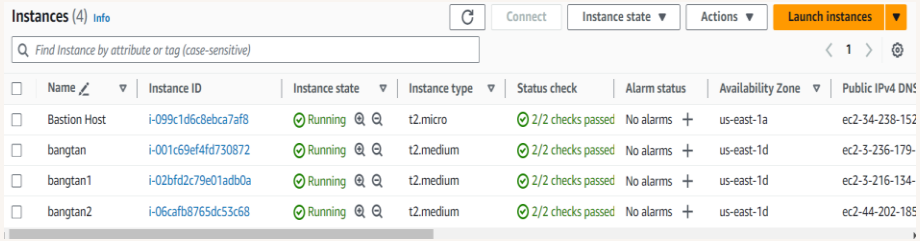
04. ADVANTAGES

1. Agentless Architecture:
2. Ease of Use
3. Idempotency
4. Scalability
5. Modularity
6. Integration
7. Open Source
8. Strong Security

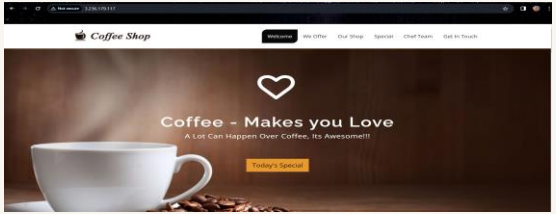
05. DEVELOPMENT



06.RESULTS




```
[exo]
bangtan1 ansible_ssh_host=3.216.134.153
bangtan2 ansible_ssh_host=44.202.185.228
```




```
ubuntu@ip-172-31-4-121:/etc/ansible$ cd /
ubuntu@ip-172-31-4-121:/$ cd ~
ubuntu@ip-172-31-4-121:~$ ansible -m ping bangtan1
The authenticity of host '3.216.134.153 (3.216.134.153)' can't be established.
ED25519 key fingerprint is SHA256:UZh2JTSnbY7uUj+WYeWM40diHoHvMpttJY40itaB/lM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
bangtan1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-4-121:~$ ansible -m ping bangtan2
The authenticity of host '44.202.185.228 (44.202.185.228)' can't be established.
ED25519 key fingerprint is SHA256:DmhkHCHAreb5Lmbbuzaf9dPe+yHJSnFJ8F8D3+DlvQc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
bangtan2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-4-121:~$
```

```
ubuntu@ip-172-31-4-121:~$ ansible-playbook rm.yml

PLAY [Install Apache HTTP Server on Ubuntu] ************************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************************************
ok: [bangtan1]
ok: [bangtan2]

TASK [Update APT package cache] ************************************************************************************************************************************************
changed: [bangtan2]
changed: [bangtan1]

TASK [Install Apache2 HTTP Server] *********************************************************************************************************************************************
changed: [bangtan1]
changed: [bangtan2]

TASK [Start and enable the Apache2 service] ************************************************************************************************************************************
ok: [bangtan1]
ok: [bangtan2]

PLAY RECAP *********************************************************************************************************************************************************************
bangtan1                   : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
bangtan2                   : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

07. CONCLUSION

- In this project, I successfully implemented an automated configuration management system using Ansible across a distributed architecture consisting of one master EC2 instance and two slave EC2 instances. By leveraging Ansible's capabilities, I created and utilized two distinct playbooks: one dedicated to the installation and configuration of an Apache server, and another for deploying and hosting a website.
- The automation of these tasks not only streamlined the setup process but also ensured consistency and reliability across the instances. This approach demonstrated the efficiency of Ansible in managing complex infrastructures, ultimately simplifying maintenance and scalability. The successful deployment and configuration of the Apache server and website illustrate the practical benefits of using Ansible for efficient system administration and application hosting.


Thank You