Ansible Dynamic Inventory:

1. Introduction

Ansible, an open-source automation tool, simplifies IT infrastructure management through automation. One of its fundamental components is the inventory system, categorizing and managing target hosts. While static inventories are common, dynamic inventories offer a more flexible and scalable approach.

2. Understanding Ansible Dynamic Inventory

2.1 Static vs. Dynamic Inventory

Static Inventory: Manually defined in files (e.g., hosts.ini), suitable for stable environments.

Dynamic Inventory: Generated dynamically through scripts or plugins, ideal for dynamic or cloud-based infrastructures.

2.2 Benefits of Dynamic Inventory

Adaptability: Accommodates frequent infrastructure changes.

Scalability: Scales seamlessly with evolving environments.

Automation: Automates inventory management, reducing manual effort.

3. Configuration

3.1 Inventory Sources

Ansible supports multiple inventory sources, including static files (hosts.ini) and dynamic inventories. To use a dynamic inventory, you need to specify the source in your Ansible configuration.

# ansible.cfg

[defaults]

inventory = ./inventory\_script.py

3.2 Inventory Scripts

Inventory scripts are executable files that Ansible runs to fetch dynamic inventory. They can be written in various languages (Python, Bash, etc.) and output JSON or INI format.

Example of a simple dynamic inventory script in Python:

#!/usr/bin/env python

import json

# Generate dynamic inventory JSON

inventory\_data = {

    'web': {

        'hosts': ['webserver1', 'webserver2'],

        'vars': {

            'ansible\_user': 'admin',

            'ansible\_port': 22

        }

    },

    'db': {

        'hosts': ['dbserver1'],

        'vars': {

            'ansible\_user': 'dbadmin',

            'ansible\_port': 3306

        }

    }

}

print(json.dumps(inventory\_data))

Specify the script path in the Ansible configuration.

Scripts must output JSON or INI format for Ansible to parse and understand.

For the above example json file will contain

{

    "web": {

        "hosts": ["webserver1", "webserver2"],

        "vars": {

            "ansible\_user": "admin",

            "ansible\_port": 22

        }

    },

    "db": {

        "hosts": ["dbserver1"],

        "vars": {

            "ansible\_user": "dbadmin",

            "ansible\_port": 3306

        }

    }

}