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

        }

    }

}

To set inventory script in aws

 Download the dynamic inventory script **ec2.py** and **ec2.ini** and put these files into your inventory directory.

wget -O <PATH OF YOUR INVENTORY DIRECTORY> <https://raw.githubusercontent.com/vshn/ansible-dynamic-inventory-ec2/master/ec2.py>

wget -O <PATH OF YOUR INVENTORY DIRECTORY> <https://raw.githubusercontent.com/vshn/ansible-dynamic-inventory-ec2/master/ec2.ini>

Now you have to provide the credentials of your AWS account. As you can provide them by exporting the access key and secret key of **IAM**user.

export AWS\_REGION='ap-south-1'  
export AWS\_ACCESS\_KEY\_ID='<YOUR ACCESS KEY>'  
export AWS\_SECRET\_ACCESS\_KEY='<YOUR SECRET ACCESS KEY>'

*Step 1:* Create a file named aws\_ec2.yaml in the inventory directory.

sudo vim aws\_ec2.yaml

Copy the following configuration to the file. If you are running an ansible server outside the AWS environment, replace add your AWS access key and secret to the config file.

---  
plugin: aws\_ec2  
aws\_access\_key: <YOUR-AWS-ACCESS-KEY-HERE>  
aws\_secret\_key: <YOUR-AWS-SECRET-KEY-HERE>  
keyed\_groups:  
 - key: tags  
 prefix: tag

If your *ansible*server is running inside the AWS environment, attach an ec2 instance role with the required AWS ec2 permissions (Mostly describe instances). This way you don’t have to add the access and secret key in the configuration. *Ansible*will automatically use the attached role to make the AWS API calls.

*Step 2:* Open /etc/ansible/ansible.cfg file.

sudo vim /etc/ansible/ansible.cfg

Find the [inventory] section or create by yourself and add the following line to enable the ec2 plugin.

enable\_plugins = aws\_ec2

It should look something like this.

[inventory]  
enable\_plugins = aws\_ec2

A computer screen shot of blue text

Description automatically generated

If you want to use the dynamic inventory as a default Ansible inventory, edit the /etc/ansible/ansible.cfg file and search for inventory parameters under defaults. Change the inventory parameter value as shown below.

inventory = /etc/ansible/inventory/aws\_ec2.yaml

A screen shot of a computer program

Description automatically generated

Now if you run the inventory list command without passing the inventory file, Ansible looks for the default location and picks up the aws\_ec2.yaml inventory file.

*Step 3:* Now let’s test the dynamic inventory configuration by listing the ec2 instances.

ansible-inventory --list

*Step 4:* Execute the following command to test if Ansible is able to ping all the machines returned by the dynamic inventory.

ansible all -m ping

So if this command run successfully than my friends you have successfully setup dynamic inventory.

Initally when I cas setting up dynamic inventory, I face lots of challenges and dificulties so I thaught to come up with a detailed article on it. So I hope that this article will help you to setup the dynamic inventroy in your server.