TrainerTests.com

Lab Exercise: Introduction to Configuration Management with Ansible for DevOps Beginners

Objective:

This lab will introduce you to the basic tasks involved in using **configuration management** tools like **Ansible**. You will learn how to install Ansible, create an inventory file, write a playbook to automate configuration tasks, and apply that configuration to one or more servers.

Step 1: Create VMs in AWS

Create Free-Tier Eligible EC2 Instances

1. Log into the AWS console and verify that the N. California region is selected
2. In the services search text box, type in **EC2**
3. Click on **EC2**
4. Click on **Launch Instances**
5. Name: `Demo`
6. Click on Ubuntu

   This is the operating system that will be installed on the EC2 instance
7. Under Instance Type, Select **t2 micro** – **Free Tier Eligible** for the instance type
8. Click **Create a new Key Pair**
9. Type `Ansible` in the **Key Pair Name** box
10. Click on **Create Key Pair**

    An EC2 Key Pair is a security credential consisting of a public key and a private key, which is used to securely access and authenticate with Amazon Elastic Compute Cloud (EC2) instances.

    *This key pair will be downloaded to your computer. **You will need this later!***
11. Under **Create security group**
    a. Check the option for **Allow HTTP Traffic from the Internet**

12. Under **Summary:**
    a. Number of Instances: **3**
13. Scroll down and click **Launch Instance**
14. Click **View all Instances**
15. Rename the instances with the following names:
    a. Web1
    b. Web2
    c. ControlVM
16. Copy and paste the Public and Private IP addresses for each VM to a text file on your computer.

Step 2: Install Ansible

Connect to the ControlVM Instance

1. The **ControlVM** Instance should be running by now. If not, take a 5 minute break.
2. Click the check box to select the **ControlVM** Instance
3. Click on **Connect**
    - Click on **EC2 Instance Connect**
    - Click on **Connect** at the bottom right
4. **Update your system**:
    - On your control node (the machine from which you will manage other servers), run the following commands to update your package list and install Ansible.

```
sudo apt install ansible -y
```

5. **Verify the installation**:
    - Run the following command to confirm that Ansible is installed:

```
ansible –version
```

Step 3: Set Up Inventory File

An **inventory file** is a file where you list the servers (or hosts) that Ansible will manage. It specifies the IP addresses or hostnames of the servers.

1. **Create the inventory file**:
   - In your home directory (or a project-specific directory), create a file called inventory:

     ```
     nano inventory
     ```

2. **Define the servers**:
   - Inside the inventory file, list the IP addresses of the servers (or VMs) you want Ansible to manage. You can group servers under categories (like web or database).

     Example:

     ```
     [web]
     192.168.1.10
     192.168.1.11

     [db]
     192.168.1.12
     ```

Step 4: Create an Ansible Playbook

An **Ansible playbook** is a YAML file where you define tasks for automating configuration management. In this lab, you'll write a simple playbook to install **Apache** on a web server.

1. **Create the playbook file**:
   - Create a new file called setup-webserver.yml:

     ```
     nano setup-webserver.yml
     ```

2. **Define the playbook**:
   - Write the following YAML code to install Apache on all web servers listed in your inventory:

```
---
- name: Install and configure Apache web servers
  hosts: web
  become: yes  # Run as root

  tasks:
    - name: Install Apache
      apt:
        name: apache2
        state: present
      when: ansible_os_family == "Debian"

    - name: Install Apache for CentOS
      yum:
        name: httpd
        state: present
      when: ansible_os_family == "RedHat"

    - name: Start and enable Apache
      service:
        name: "{{ 'apache2' if ansible_os_family == 'Debian' else
'httpd' }}"
        state: started
        enabled: yes
```

3. **Save the playbook** and exit the editor.

Step 5: Copy Key Pairs

1. Launch Powershell on Your Computer
2. Issue the following command:

   NOTE: Substitute the path shown for the path where your PEM file was downloaded on your computer. Substitute the IP address for the public IP of your ControlVM.

   ```
   scp -i C:\Users\Rick\Downloads\ansible.pem C:\Users\Rick\Downloads\ansible.pem ubuntu@3.101.33.2:/home/ubuntu/
   ```

3. Return to the console of the Control VM
   - Issue this command to change permissions on the PEM file:

```
chmod 400 /home/ubuntu/ansible.pem
```

- Issue this command to test SSH access from the control VM to one of your web servers.
- Be sure to replace the IP shown with the **private** IP of one of your web server instances.

  ```
  ssh -i /home/ubuntu/ansible.pem ubuntu@172.31.25.137
  ```

- Type exit to end the SSH session to the web server.

Step 6: Run the Ansible Playbook

4. **Execute the playbook**:
   - Use the ansible-playbook command to run the playbook against the servers in the web group:

     ```
     ansible-playbook -i inventory setup-webserver.yml --private-key
     /home/ubuntu/ansible.pem
     ```

5. **Monitor the output**:
   - Ansible will show the output of each task as it runs. You should see a summary indicating whether the Apache package was installed, started, and enabled.

Step 7: Verify the Configuration

1. **Check if Apache is running**:
   - After the playbook runs, SSH into one of the web servers to verify that Apache has been installed and is running:

     ```
     ssh ubuntu@192.168.1.10 -i ~/.ssh/your-key.pem
     sudo systemctl status apache2  # Use httpd for CentOS
     ```

2. **Access the web server**:
   - Open a browser and enter the **Public** IP address of one of the web servers (e.g., http://1.2.3.4). You should see the default Apache welcome page.

Step 8: Clean Up

1. To avoid charges on cloud platforms or keep your local environment clean, you can remove the VMs or the services installed.
    - **Stop or terminate the VMs** in your cloud provider dashboard or use the shutdown command on each VM.
    - **Remove the Apache service** by modifying the playbook or manually using the package manager.