ANSIBLE

Ansible is a suite of software tools that enables infrastructure as code. It is open-source and the suite includes software provisioning, configuration management, and application deployment functionality. One of its standout features is its agentless architecture, which eliminates the need for installing additional software agents on target systems. Instead, Ansible leverages secure communication protocols like SSH to manage remote hosts. This approach enhances the tool's ease of use and deployment, making it a popular choice for automation tasks.

Normally, there is a control node and host nodes. Ansible is installed in the control node and will execute A basic Ansible environment has three main components: control node, managed node, and inventory. Ansible is installed in the control node and will execute the ansible playbook to deploy in managed nodes using the inventory file in the control node that describes the managed nodes to Ansible.

Working

The working of Ansible revolves around the automation of tasks and the management of infrastructure using a declarative and idempotent approach. Here's how Ansible works step by step:

Inventory Configuration:

- An Ansible project begins with an inventory file that lists the hosts or servers you want to manage.
- The inventory file can be a static text file listing hostnames or IP addresses, or a dynamic file generated by external scripts or cloud providers.

Playbook Creation:

- In Ansible, you define the desired state of the systems in playbooks.
- Playbooks are written in YAML and consist of tasks that describe the actions you want to perform on the hosts.
- Playbooks can include roles and variables, allowing for modularity and reusability.

Task Execution:

- When you run an Ansible playbook, Ansible connects to the target hosts via SSH (for Linux/Unix systems).
- It identifies the tasks that need to be executed on each host based on the playbook's content.

Module Execution:

- Ansible uses modules to perform tasks. Modules are small units of code that Ansible executes on remote hosts.
- Each task in a playbook corresponds to a module. Modules are idempotent, meaning they only make changes if necessary.

Desired State Implementation:

- Ansible checks the current state of the system against the desired state defined in the playbook.
- If the current state differs from the desired state, Ansible executes the necessary tasks to bring the system into compliance with the playbook.

Idempotent Operations:

- One of Ansible's core principles is idempotence. This means you can run the same playbook multiple times without causing unexpected changes.
- If a task has already been executed and the system is in the desired state, Ansible takes no action.

Assignment

This guide provides a step-by-step walkthrough of the process to install Ansible, create an EC2 instance on AWS, and use Ansible playbooks to automate the installation, starting, and stopping of an Apache web server on the EC2 instance. This documentation is particularly useful for software engineers and system administrators.

Prerequisites:

- An AWS account with access to EC2 instances.
- A running EC2 instance with SSH access.

Steps:

1. **Install Ansible**: Begin by installing Ansible on your local machine. Follow the official Ansible installation guide for instructions tailored to your specific operating system.

2. **Resolve Locale Encoding Error:**
   export LC_ALL=C.UTF-8
   export LANG=C.UTF-8

3. **Verify Ansible Installation:**
   ansible --version

4. **Create an EC2 Instance:** In your aws account, create an ec2 instance. Make sure you have the necessary access permissions, an appropriate security group, and a key pair for ssh access.

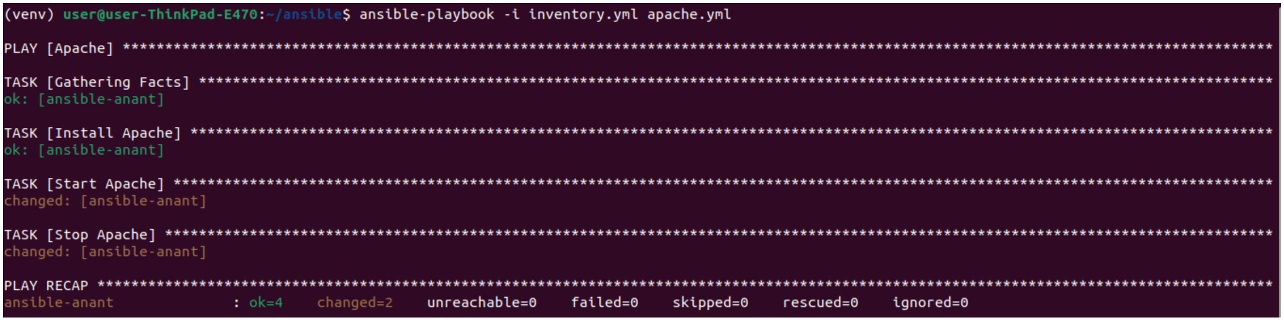
5. **Create an Ansible Inventory File:** Create an Ansible inventory file, e.g., **inventory.yml**, and configure your EC2 instance. Define the instance's IP address and the SSH user for connecting to it.

6. **Create an Ansible Playbook:** Develop an Ansible playbook, e.g., **apache.yml**, to describe the tasks for installing and managing Apache on your EC2 instance.

7. **Run the Ansible Playbook:**

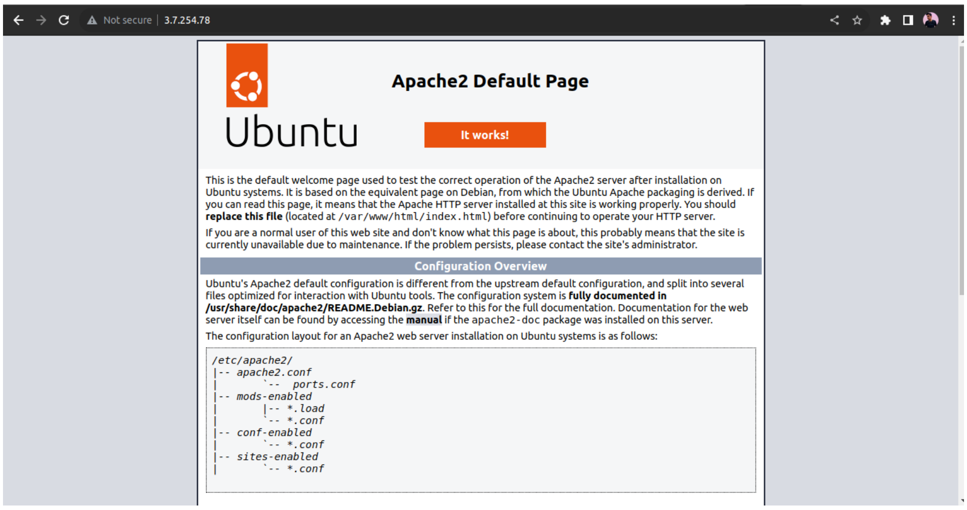
ansible-playbook -i inventory.yml your_playbook.yml

Output



```
(venv) user@user-ThinkPad-E470:~/ansible$ ansible-playbook -i inventory.yml apache.yml

PLAY [Apache] ***************************************************************************************************************************

TASK [Gathering Facts] ******************************************************************************************************************
ok: [ansible-anant]

TASK [Install Apache] *******************************************************************************************************************
ok: [ansible-anant]

TASK [Start Apache] *********************************************************************************************************************
changed: [ansible-anant]

TASK [Stop Apache] **********************************************************************************************************************
changed: [ansible-anant]

PLAY RECAP ******************************************************************************************************************************
ansible-anant              : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

To recheck whether the apache server started or not, you can just enter the ip address of the host in the web browser, you can see the below page.



Apache2 Default Page

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

References:

Official Documentation: https://docs.ansible.com/ansible/latest/index.html
Blog: https://linuxopsys.com/topics/ansible-playbook-to-install-apache
Blog: https://www.freecodecamp.org/news/what-is-ansible/