

Lesson 01 Demo 01

Setting Up LAMP Stack Using Ansible

Objective: To set up and configure LAMP Stack dependencies with Ansible to pre-configure the infrastructure for new developers and automate the configuration management for the same

Tools required: Ubuntu OS

Prerequisites: You need to have Ansible installed.

Steps to be followed:

1. Configure the Ansible playbook
2. Execute the LAMP playbook

Step 1: Configure the Ansible playbook

1.1 Execute the command given below to check if Ansible is installed:

ansible --version

```
shreemayeebhatt@ip-172-31-25-47:~$ ansible --version
ansible [core 2.12.10]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/shreemayeebhatt/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/shreemayeebhatt/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0]
  jinja version = 3.0.3
  libyaml = True
```

1.2 Execute the following commands to navigate to the **hosts** directory:

cd /etc/ansible

nano hosts

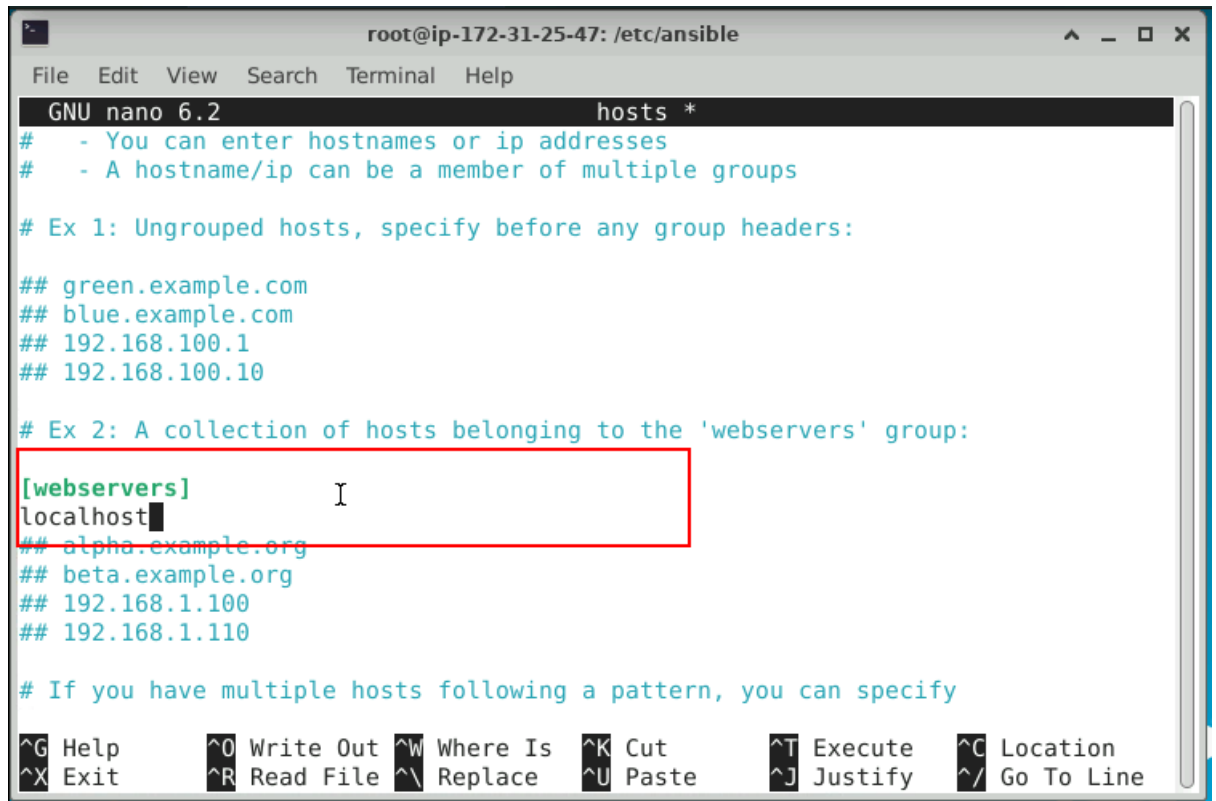
```
root@ip-172-31-25-47:/home/shreemayeebhatt# cd /etc/ansible
root@ip-172-31-25-47:/etc/ansible# nano hosts
```

Note: You may need to use the commands with **sudo** to change into the root user.

1.3 Configure the following **[webserver]** in the **hosts** file and save it:

[webserver]

localhost



```
root@ip-172-31-25-47: /etc/ansible
GNU nano 6.2 hosts *
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers:

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webserver' group:
[webserver]
localhost
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern, you can specify

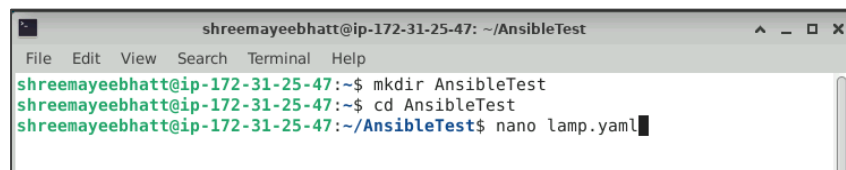
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

1.4 Create an Ansible playbook with the command given below:

mkdir AnsibleTest

cd AnsibleTest

nano lamp.yaml



```
shreemayeebhatt@ip-172-31-25-47: ~/AnsibleTest
File Edit View Search Terminal Help
shreemayeebhatt@ip-172-31-25-47:~$ mkdir AnsibleTest
shreemayeebhatt@ip-172-31-25-47:~$ cd AnsibleTest
shreemayeebhatt@ip-172-31-25-47:~/AnsibleTest$ nano lamp.yaml
```

1.5 Enter the following YAML script in the **lamp.yaml** file to set up a LAMP stack (Linux, Apache, MySQL, PHP) on an Ubuntu server:

- name: Install LAMP stack on Ubuntu

hosts: lamp_server

become: yes

vars:

mysql_root_password: "your_mysql_root_password"

php_packages:

- php
- php-mysql
- libapache2-mod-php
- php-cli
- php-curl
- php-json
- php-mbstring
- php-xml
- php-zip

tasks:

- name: Update apt package index

apt:

update_cache: yes

- name: Install Apache

apt:

name: apache2

state: present

- name: Start and enable Apache service

service:

name: apache2

state: started

enabled: yes

- name: Install MySQL

debconf:

name: "mysql-server"

question: "mysql-server/root_password"

value: "{{ mysql_root_password }}"

vtype: "password"

- name: Install MySQL again to set the root password

debconf:

name: "mysql-server"

question: "mysql-server/root_password_again"

value: "{{ mysql_root_password }}"

vtype: "password"

- name: Install MySQL server

apt:

name: mysql-server

state: present

- name: Start and enable MySQL service

service:

name: mysql

state: started

enabled: yes

- name: Install PHP and related packages

apt:

name: "{{ php_packages }}"

state: present

- name: Create info.php to test PHP

copy:

dest: /var/www/html/info.php

content: |

<?php

phpinfo();

?>

- name: Restart Apache to apply PHP installation

service:

name: apache2

state: restarted

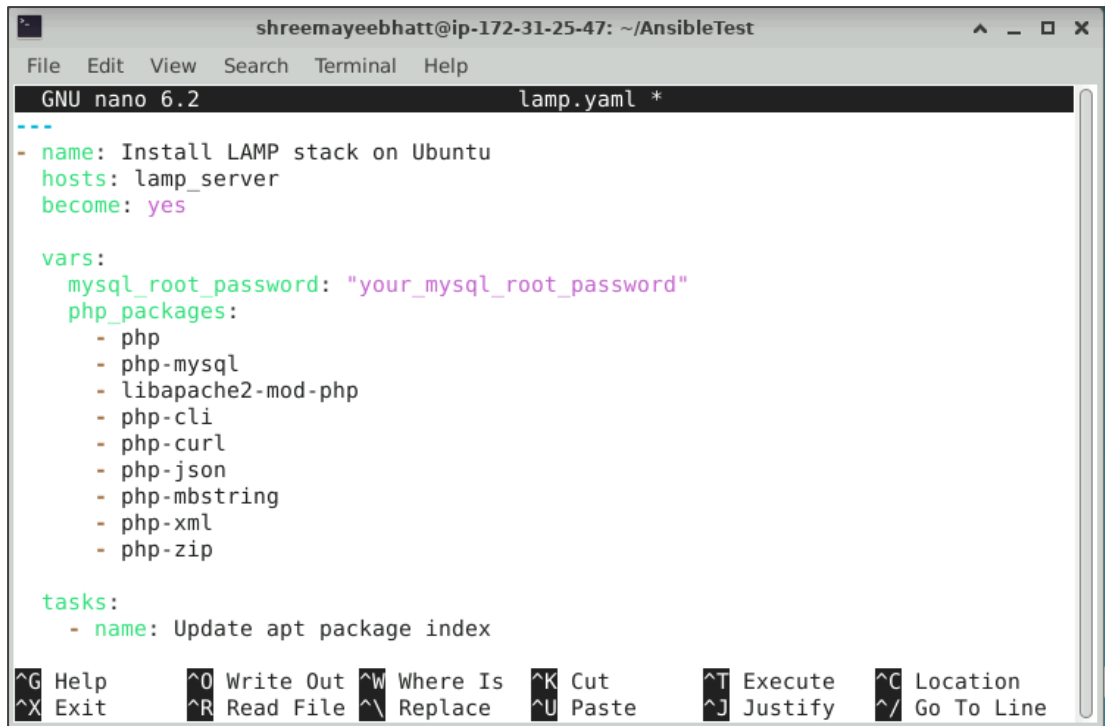
handlers:

- name: Restart Apache

service:

name: apache2

state: restarted



```
shreemayeebhatt@ip-172-31-25-47: ~/AnsibleTest
File Edit View Search Terminal Help
GNU nano 6.2 lamp.yaml *
---
- name: Install LAMP stack on Ubuntu
  hosts: lamp_server
  become: yes

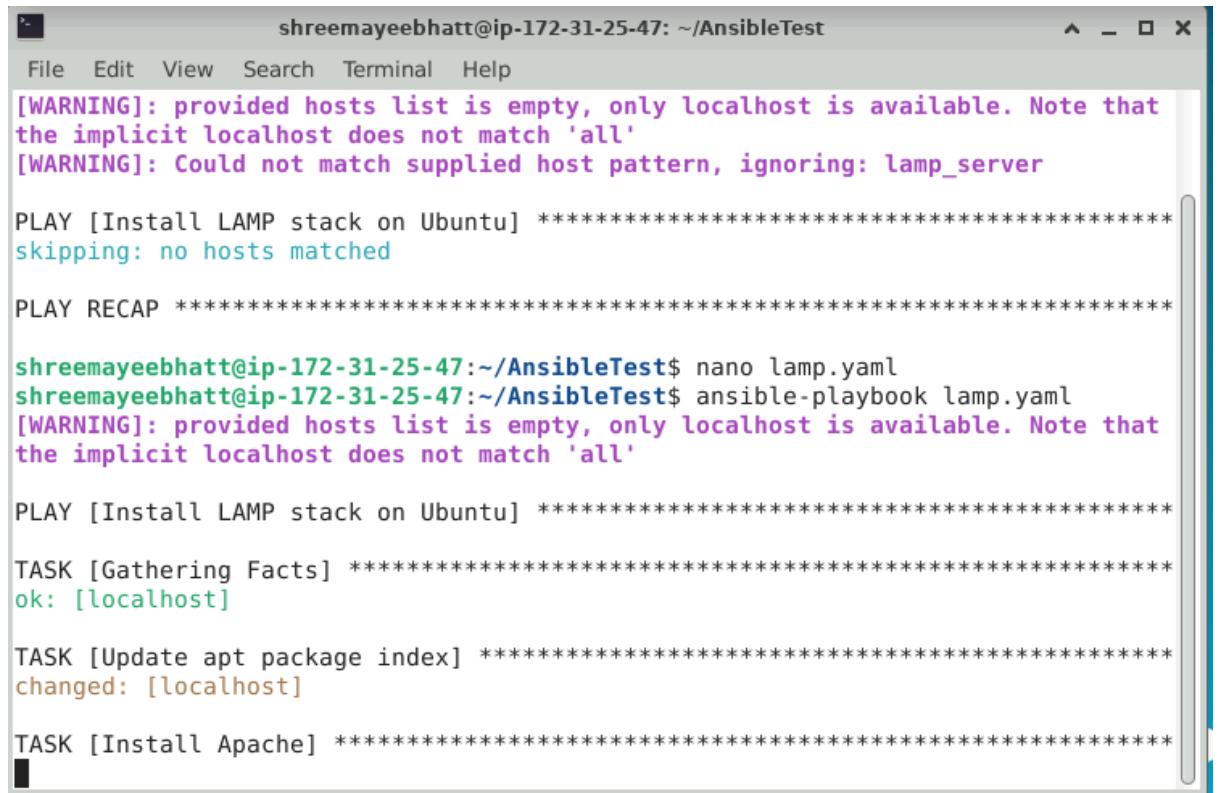
  vars:
    mysql_root_password: "your_mysql_root_password"
    php_packages:
      - php
      - php-mysql
      - libapache2-mod-php
      - php-cli
      - php-curl
      - php-json
      - php-mbstring
      - php-xml
      - php-zip

  tasks:
    - name: Update apt package index
```

Step 2: Execute the LAMP playbook

2.1 Execute the following command to run the playbook:

ansible-playbook lamp.yaml



```
shreemayeebhatt@ip-172-31-25-47: ~/AnsibleTest
File Edit View Search Terminal Help

[WARNING]: provided hosts list is empty, only localhost is available. Note that
the implicit localhost does not match 'all'
[WARNING]: Could not match supplied host pattern, ignoring: lamp_server

PLAY [Install LAMP stack on Ubuntu] *****
skipping: no hosts matched

PLAY RECAP *****

shreemayeebhatt@ip-172-31-25-47:~/AnsibleTest$ nano lamp.yaml
shreemayeebhatt@ip-172-31-25-47:~/AnsibleTest$ ansible-playbook lamp.yaml
[WARNING]: provided hosts list is empty, only localhost is available. Note that
the implicit localhost does not match 'all'

PLAY [Install LAMP stack on Ubuntu] *****

TASK [Gathering Facts] *****
ok: [localhost]

TASK [Update apt package index] *****
changed: [localhost]

TASK [Install Apache] *****
```

By following these steps, you have successfully set up and configured LAMP Stack dependencies with Ansible to pre-configure the infrastructure for new developers and automate the configuration management.

Note: In the real-world scenario, you will have a list of hosts where you will do the configuration management. This will be discussed in the upcoming lessons.