Deployment of WordPress Environment using ansible playbook:

Introduction:

Ansible:

Ansible is a configuration and orchestration management tool where applications are deployed automatically in a variety of environments.

Installing Ansible

Now, let's install Ansible:

Updating Control Node by using following command:

sudo apt-get update

Commands to install ansible in ubuntu:

sudo apt-get install software-properties-common

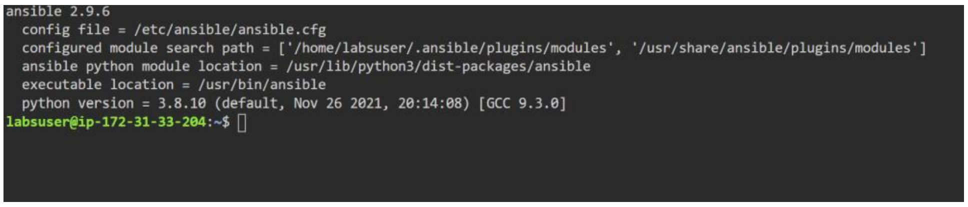
sudo apt-add-repository ppa:ansible/ansible

sudo apt-get update

sudo apt-get install ansible

ansible --version

fig1: showing ansible version installed in the machine: -



```
ansible 2.9.6
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/labsuser/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.8.10 (default, Nov 26 2021, 20:14:08) [GCC 9.3.0]
labsuser@ip-172-31-33-204:~$ 
```

Configure Ansible on controller and managed node

After installing ansible I setup Ansible environment and configure my controller and managed nodes.

Create normal user: -

I created a user called ans by using following commands:

- useradd ans

Assign a password to this user:

- passwd ans

I Repeated the same steps on managed nodes i.e., created the same user on managed nodes

Configure password less authentication: -

I set up password less authentication for on ans user from controller to all the managed nodes. This is to ensure that the controller can connect to all the managed nodes without any password prompt.

Login as ans user on the controller node and generate private public key pair using ssh-keygen.

step to create private and public key pair inside ~/.ssh

I used ssh-copy-id to copy the keys to remote managed server and add it to authorized keys.

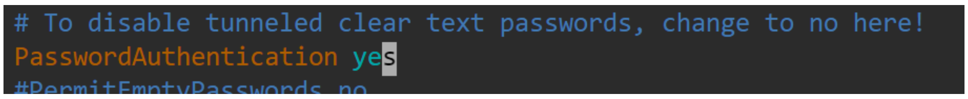
ssh-copy-id ans@172.31.36.166

ssh ans@172.31.36.166

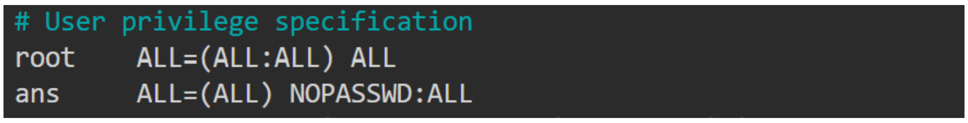
Configure privilege escalation using sudo:

ans user would need privilege escalation I added ans user on sudoers file and I edited sshd_config file

Change password authentication no to yes



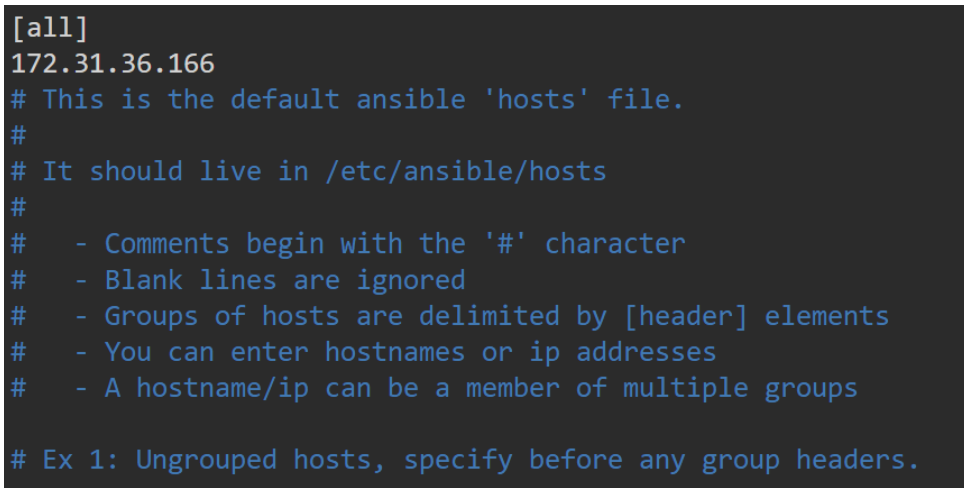
```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
```



```
# User privilege specification
root    ALL=(ALL:ALL) ALL
ans     ALL=(ALL) NOPASSWD:ALL
```

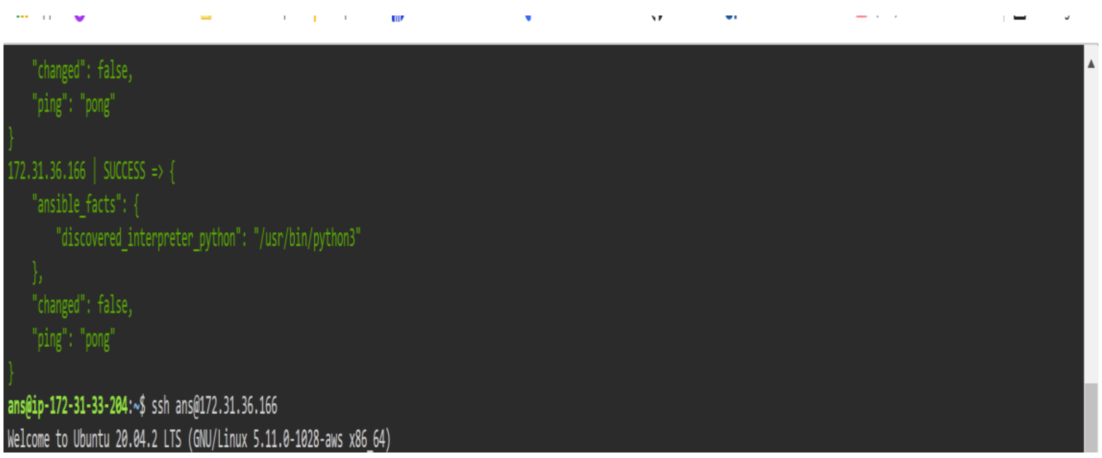
Verify ansible connectivity: -

Next, I verify the connection between control and managed nodes. To achieve I added ip address of managed nodes in /etc/ansible/hosts.it is default inventory file



```
[all]
172.31.36.166
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - 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.
```

Next, I try to ping your managed nodes using the controller node as ans user with ansible all -m ping command. following fig2 shows



```
    "changed": false,
    "ping": "pong"
}
172.31.36.166 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ans@ip-172-31-33-204:~$ ssh ans@172.31.36.166
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.11.0-1028-aws x86_64)
```

Deployment of WordPress Environment using ansible playbook:

ansible playbook: -

Playbooks are the files where Ansible code is written. Playbooks are written in YAML format. YAML stands for Yet Another Markup Language. Playbooks are one of the core features of Ansible and tell Ansible what to execute. They are like a to-do list for Ansible that contains a list of tasks

In order to install WordPress on managed node we need to write a playbook

Iam going to create a ansible role for WordPress installation

First, we need to create a role where to install WordPress:

Steps to create ansible role using following command

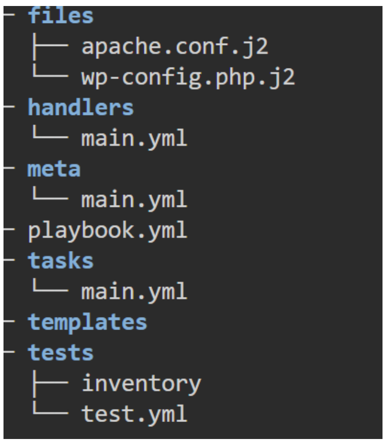
ansible-galaxy init wordpress_ans

it will show message below

Role wordpress_ans was created successfully

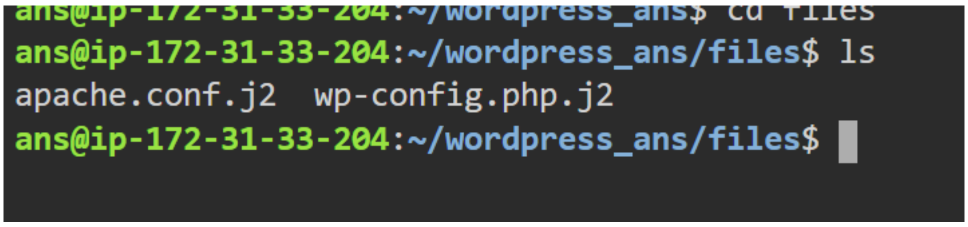
I used tree command to list the ansible role directory structure

tree wordpress_ans .the foll fig shows the directory created



```
- files
  ├── apache.conf.j2
  └── wp-config.php.j2
- handlers
  └── main.yml
- meta
  └── main.yml
- playbook.yml
- tasks
  └── main.yml
- templates
- tests
  ├── inventory
  └── test.yml
```

cd file and create two templates' files



```
ans@ip-172-31-33-204:~/wordpress_ans$ cd files
ans@ip-172-31-33-204:~/wordpress_ans/files$ ls
apache.conf.j2  wp-config.php.j2
ans@ip-172-31-33-204:~/wordpress_ans/files$ 
```

Files/apache.conf.j2: Template file for setting up the Apache Virtual Host.(The apache.conf. j2 file is a Jinja 2 template file that configures a new Apache Virtual Host. The variables used within this template are defined in the vars/default.yml variable file.)

apache.conf.j2:

```
<VirtualHost *:{{ http_port }}>
   ServerAdmin webmaster@localhost
   ServerName {{ http_host }}
   ServerAlias www.{{ http_host }}
   DocumentRoot /var/www/{{ http_host }}/wordpress
   ErrorLog ${APACHE_LOG_DIR}/error.log
   CustomLog ${APACHE_LOG_DIR}/access.log combined

   <Directory /var/www/{{ http_host }}>
        Options -Indexes
        AllowOverride All
   </Directory>

   <IfModule mod_dir.c>
      DirectoryIndex index.php index.html index.cgi index.pl  index.xhtml index.htm
   </IfModule>

</VirtualHost>
```

----

files/wp-config.php. j2: Template file for setting up WordPress's configuration file. (The wp-config.php.j2 file is another Jinja template, used to set up the main configuration file used by WordPress. The variables used within this template are defined in the vars/default.yml variable file. Unique authentication keys and salts are generated using a hash function.)

wp-config.php. j2:

```
<?php
```

```
/**
* The base configuration for WordPress
*
* The wp-config.php creation script uses this file during the
* installation. You don't have to use the web site, you can
* copy this file to "wp-config.php" and fill in the values.
*
* This file contains the following configurations:
*
* * MySQL settings
* * Secret keys
* * Database table prefix
* * ABSPATH
*
* @link https://codex.wordpress.org/Editing_wp-config.php
*
* @package WordPress
*/

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', '{{ mysql_db }}' );

/** MySQL database username */
define( 'DB_USER', '{{ mysql_user }}' );

/** MySQL database password */
define( 'DB_PASSWORD', '{{ mysql_password }}' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );
```

```
/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );


/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );


/** Filesystem access **/
define('FS_METHOD', 'direct');


/**#@+
* Authentication Unique Keys and Salts.
*
* Change these to different unique phrases!
* You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}
* You can change these at any point in time to invalidate all existing cookies. This will force all users to have to log in again.
*
* @since 2.6.0
*/
define( 'AUTH_KEY',         '{{ lookup('password', '/dev/null chars=ascii_letters length=64') }}' );
define( 'SECURE_AUTH_KEY',  '{{ lookup('password', '/dev/null chars=ascii_letters length=64') }}' );
define( 'LOGGED_IN_KEY',    '{{ lookup('password', '/dev/null chars=ascii_letters length=64') }}' );
define( 'NONCE_KEY',        '{{ lookup('password', '/dev/null chars=ascii_letters length=64') }}' );
define( 'AUTH_SALT',        '{{ lookup('password', '/dev/null chars=ascii_letters length=64') }}' );
define( 'SECURE_AUTH_SALT', '{{ lookup('password', '/dev/null chars=ascii_letters length=64') }}' );
define( 'LOGGED_IN_SALT',   '{{ lookup('password', '/dev/null chars=ascii_letters length=64') }}' );
define( 'NONCE_SALT',       '{{ lookup('password', '/dev/null chars=ascii_letters length=64') }}' );


/**#@-*/


/**
```

```
* WordPress Database Table prefix.
*
* You can have multiple installations in one database if you give each
* a unique prefix. Only numbers, letters, and underscores please!
*/
$table_prefix = 'wp_';

/**
* For developers: WordPress debugging mode.
*
* Change this to true to enable the display of notices during development.
* It is strongly recommended that plugin and theme developers use WP_DEBUG
* in their development environments.
*
* For information on other constants that can be used for debugging,
* visit the Codex.
*
* @link https://codex.wordpress.org/Debugging_in_WordPress
*/
define( 'WP_DEBUG', false );

/* That's all, stop editing! Happy publishing. */

/** Absolute path to the WordPress directory. */
if ( ! defined( 'ABSPATH' ) ) {
define( 'ABSPATH', dirname( __FILE__ ) . '/' );
}

/** Sets up WordPress vars and included files. */
require_once( ABSPATH . 'wp-settings.php' );
```

Next i moved to vars folder using below command

cd vars

And i created default.yml playbook (to Variable file for customizing playbook settings)

The default.yml variable file contains values that will be used within the playbook tasks, such as the database settings and the domain name to configure within Apache.

default.yaml

```
---

#System Settings

php_modules: [ 'php-curl', 'php-gd', 'php-mbstring', 'php-xml', 'php-xmlrpc', 'php-soap', 'php-intl', 'php-zip' ]


#MySQL Settings

mysql_root_password: "mysql_root_password"

mysql_db: "wordpress"

mysql_user: "vani"

mysql_password: "password"


#HTTP Settings

http_host: "your_domain"

http_conf: "your_domain.conf"

http_port: "80"
```

move to wordpress _ans

Playbook to install wordpress (playbook.yml)

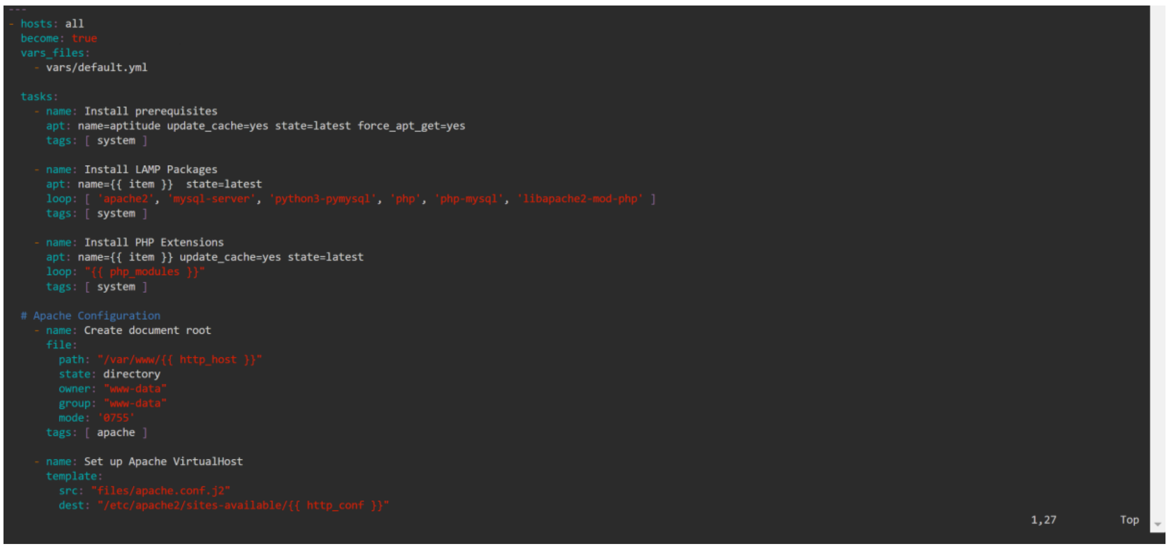
This playbook.yml: The playbook file, containing the tasks to be executed on the remote server.The playbook.yml file is where all tasks from this setup are defined. It starts by defining the group of servers that should be the target of this setup (new), after which it uses become: true to define that tasks should be executed with privilege escalation (sudo) by default. Then, it includes the vars/default.yml variable file to load configuration options.

Playbook.yml

```
---

- hosts:  all

  become: true

  vars_files:

    - vars/default.yml
```

```
tasks:
  - name: Install prerequisites
    apt: name=aptitude  state=latest force_apt_get=yes
    tags: [ system ]

  - name: Install LAMP Packages
    apt: name={{ item }} update_cache=yes state=latest
    loop: [ 'apache2', 'mysql-server', 'python3-pymysql', 'php', 'php-mysql', 'libapache2-mod-php' ]
    tags: [ system ]

  - name: Install PHP Extensions
    apt: name={{ item }}  state=latest
    loop: "{{ php_modules }}"
    tags: [ system ]

# Apache Configuration
  - name: Create document root
    file:
      path: "/var/www/{{ http_host }}"
      state: directory
      owner: "www-data"
      group: "www-data"
      mode: '0755'
    tags: [ apache ]

  - name: Set up Apache VirtualHost
    template:
      src: "files/apache.conf.j2"
      dest: "/etc/apache2/sites-available/{{ http_conf }}"
    notify: Reload Apache
    tags: [ apache ]
  - name: Enable rewrite module
    shell: /usr/sbin/a2enmod rewrite
    notify: Reload Apache
    tags: [ apache ]

  - name: Enable new site
    shell: /usr/sbin/a2ensite {{ http_conf }}
    notify: Reload Apache
    tags: [ apache ]

  - name: Disable default Apache site
    shell: /usr/sbin/a2dissite 000-default.conf
    notify: Restart Apache
    tags: [ apache ]

# MySQL Configuration
  - name: Set the root password
    mysql_user:
      name: root
      host_all: yes
      password: "{{mysql_root_password}}"
      login_unix_socket: /var/run/mysqld/mysqld.sock
    tags: [ mysql, mysql-root]


  - name: Remove all anonymous user accounts
    mysql_user:
      name: ''
      host_all: yes
      state: absent
      login_user: root
      login_password: "{{ mysql_root_password }}"
    tags: [ mysql ]

  - name: Remove the MySQL test database
    mysql_db:
      name: test
      state: absent
      login_user: root
      login_password: "{{ mysql_root_password }}"
    tags: [ mysql ]

  - name: Creates database for WordPress
    mysql_db:
      name: "{{ mysql_db }}"
      state: present
      login_user: vani
      login_password: "{{ mysql_root_password }}"
    tags: [ mysql ]

  - name: Create MySQL user for WordPress
    mysql_user:
      name: "{{ mysql_user }}"
      password: "{{ mysql_password }}"
      priv: "{{ mysql_db }}.*:ALL"
      state: present
      login_user: root
      login_password: "{{ mysql_root_password }}"
    tags: [ mysql ]

# UFW Configuration
  - name: "UFW - Allow HTTP on port {{ http_port }}"
    ufw:
        rule: allow
        port: "{{ http_port }}"
        proto: tcp
      tags: [ system ]

# WordPress Configuration
    - name: Download and unpack latest WordPress
      unarchive:
        src: https://wordpress.org/latest.tar.gz
        dest: "/var/www/{{ http_host }}"
        remote_src: yes
        creates: "/var/www/{{ http_host }}/wordpress"
      tags: [ wordpress ]

    - name: Set ownership
      file:
       path: "/var/www/{{ http_host }}"
       state: directory
       recurse: yes
       owner: www-data
       group: www-data
      tags: [ wordpress ]

    - name: Set permissions for directories
      shell: "/usr/bin/find /var/www/{{ http_host }}/wordpress/ -type d -exec chmod 750 {} \\;"
      tags: [ wordpress ]

    - name: Set permissions for files
      shell: "/usr/bin/find /var/www/{{ http_host }}/wordpress/ -type f -exec chmod 640 {} \\;"
      tags: [ wordpress ]

    - name: Set up wp-config
```

```
    template:
      src: "files/wp-config.php.j2"
      dest: "/var/www/{{ http_host }}/wordpress/wp-config.php"
    tags: [ wordpress ]

handlers:
  - name: Reload Apache
    service:
      name: apache2
      state: reloaded

  - name: Restart Apache
    service:
      name: apache2
      state: restarted.
```

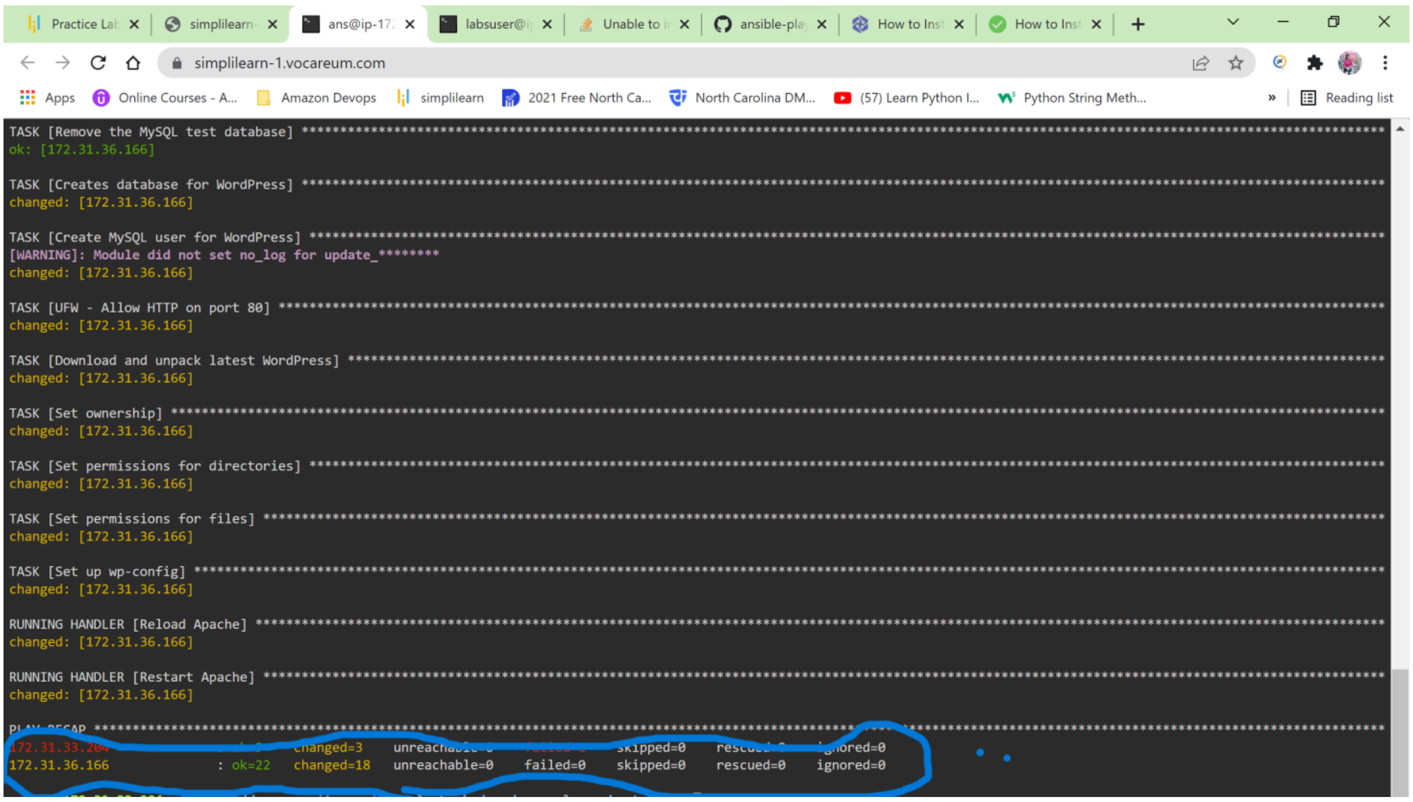


```
---
- hosts: all
  become: true
  vars_files:
    - vars/default.yml

  tasks:
    - name: Install prerequisites
      apt: name=aptitude update_cache=yes state=latest force_apt_get=yes
      tags: [ system ]

    - name: Install LAMP Packages
      apt: name={{ item }}  state=latest
      loop: [ 'apache2', 'mysql-server', 'python3-pymysql', 'php', 'php-mysql', 'libapache2-mod-php' ]
      tags: [ system ]

    - name: Install PHP Extensions
      apt: name={{ item }} update_cache=yes state=latest
      loop: "{{ php_modules }}"
      tags: [ system ]

  # Apache Configuration
    - name: Create document root
      file:
        path: "/var/www/{{ http_host }}"
        state: directory
        owner: "www-data"
        group: "www-data"
        mode: '0755'
      tags: [ apache ]

    - name: Set up Apache VirtualHost
      template:
        src: "files/apache.conf.j2"
        dest: "/etc/apache2/sites-available/{{ http_conf }}"
                                                                        1,27          Top
```

===

Running ansible playbook by using following command

ansible-playbook playbook.yml.once I run the playbook, I got following output shown in the below fig.



```
TASK [Remove the MySQL test database] ******************************************************************************
ok: [172.31.36.166]

TASK [Creates database for WordPress] ******************************************************************************
changed: [172.31.36.166]

TASK [Create MySQL user for WordPress] *****************************************************************************
[WARNING]: Module did not set no_log for update_********
changed: [172.31.36.166]

TASK [UFW - Allow HTTP on port 80] *********************************************************************************
changed: [172.31.36.166]

TASK [Download and unpack latest WordPress] ************************************************************************
changed: [172.31.36.166]

TASK [Set ownership] ***********************************************************************************************
changed: [172.31.36.166]

TASK [Set permissions for directories] *****************************************************************************
changed: [172.31.36.166]

TASK [Set permissions for files] ***********************************************************************************
changed: [172.31.36.166]

TASK [Set up wp-config] ********************************************************************************************
changed: [172.31.36.166]

RUNNING HANDLER [Reload Apache] ************************************************************************************
changed: [172.31.36.166]

RUNNING HANDLER [Restart Apache] ***********************************************************************************
changed: [172.31.36.166]

PLAY RECAP *********************************************************************************************************
172.31.33.204              : ok=[illegible]    changed=3    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
172.31.36.166              : ok=22   changed=18   unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

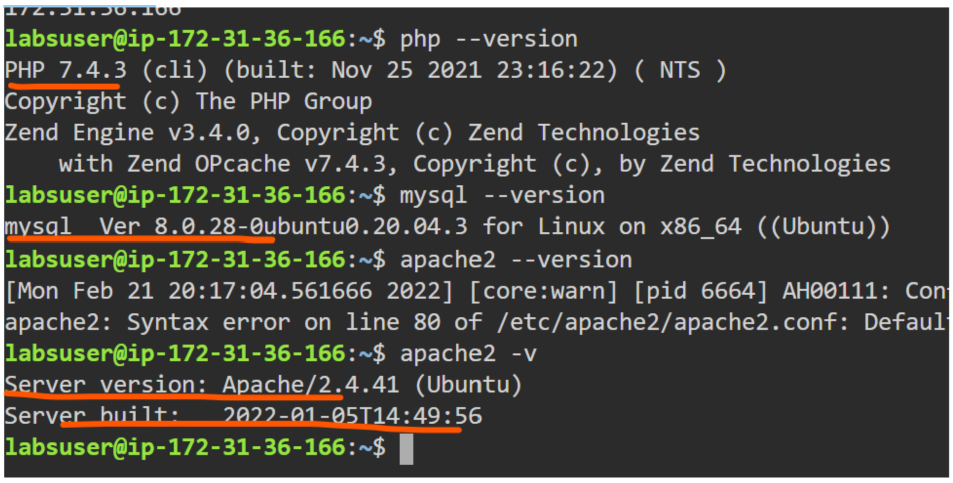
When the playbook is finished running, i go to your web browser to finish WordPress's installation from there.

and also verify in managed node using following command:

php --version

mysql --version

apache2 -v

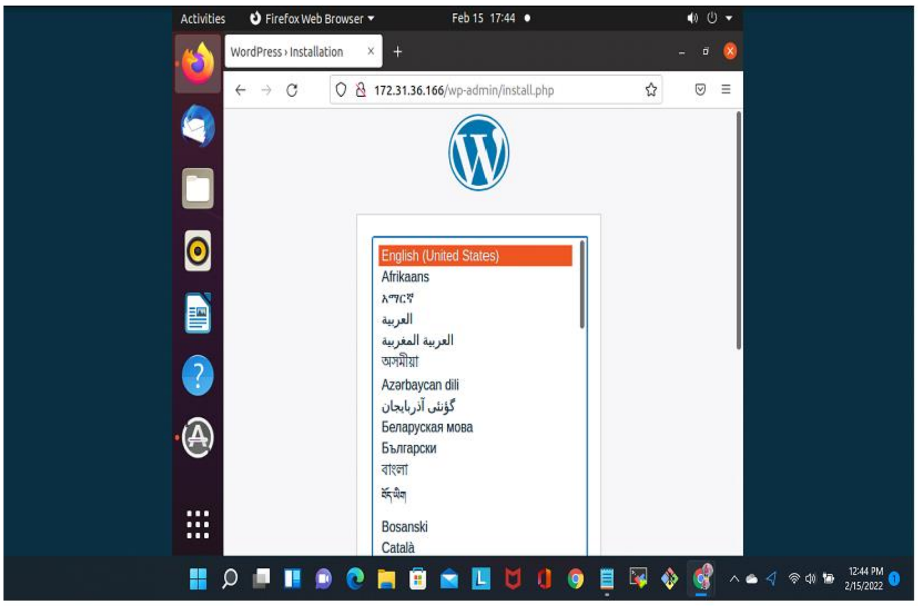


```
labsuser@ip-172-31-36-166:~$ php --version
PHP 7.4.3 (cli) (built: Nov 25 2021 23:16:22) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
labsuser@ip-172-31-36-166:~$ mysql --version
mysql  Ver 8.0.28-0ubuntu0.20.04.3 for Linux on x86_64 ((Ubuntu))
labsuser@ip-172-31-36-166:~$ apache2 --version
[Mon Feb 21 20:17:04.561666 2022] [core:warn] [pid 6664] AH00111: Con
apache2: Syntax error on line 80 of /etc/apache2/apache2.conf: Defaul
labsuser@ip-172-31-36-166:~$ apache2 -v
Server version: Apache/2.4.41 (Ubuntu)
Server built:   2022-01-05T14:49:56
labsuser@ip-172-31-36-166:~$
```

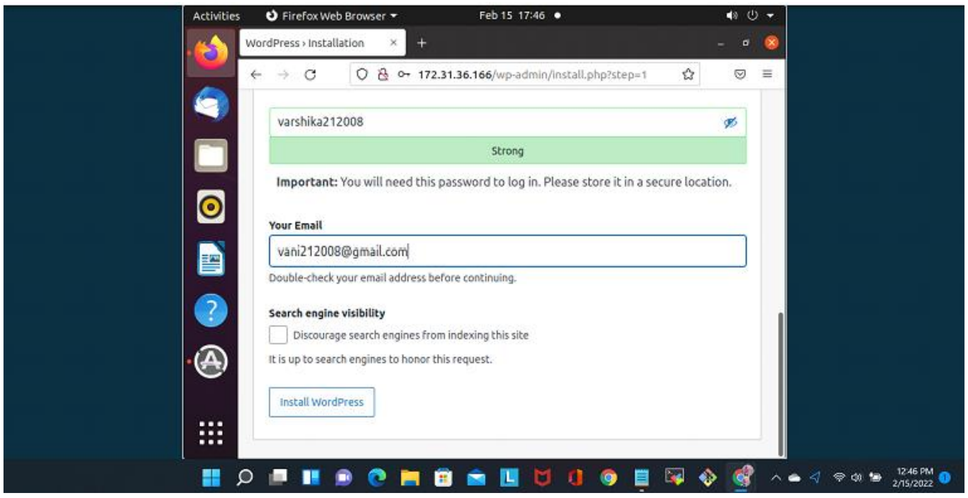
Navigate server's domain name or public IP address:

http://172.31.36.166:80

Its shows a page like this in below figures:



After selecting the language I click the install wordprss button and its taken me into the l WordPress user and password page and i logged into my control panel



varshika212008

Strong

Important: You will need this password to log in. Please store it in a secure location.

Your Email

vani212008@gmail.com

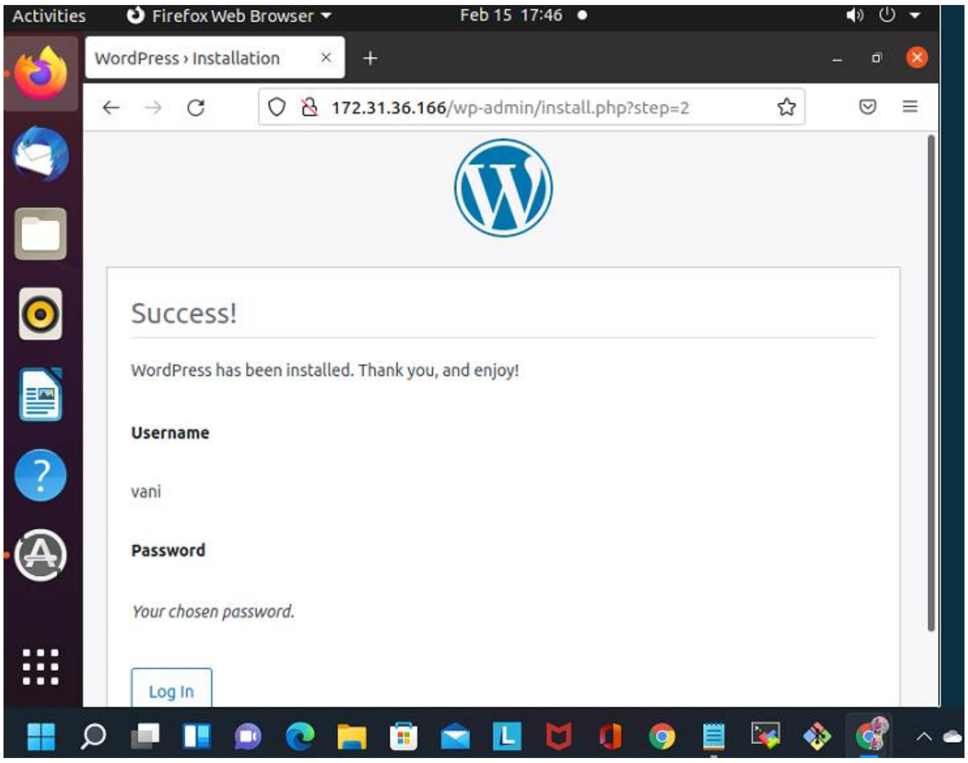
Double-check your email address before continuing.

Search engine visibility

☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

It taken to a page that prompts you me to log in



Success!

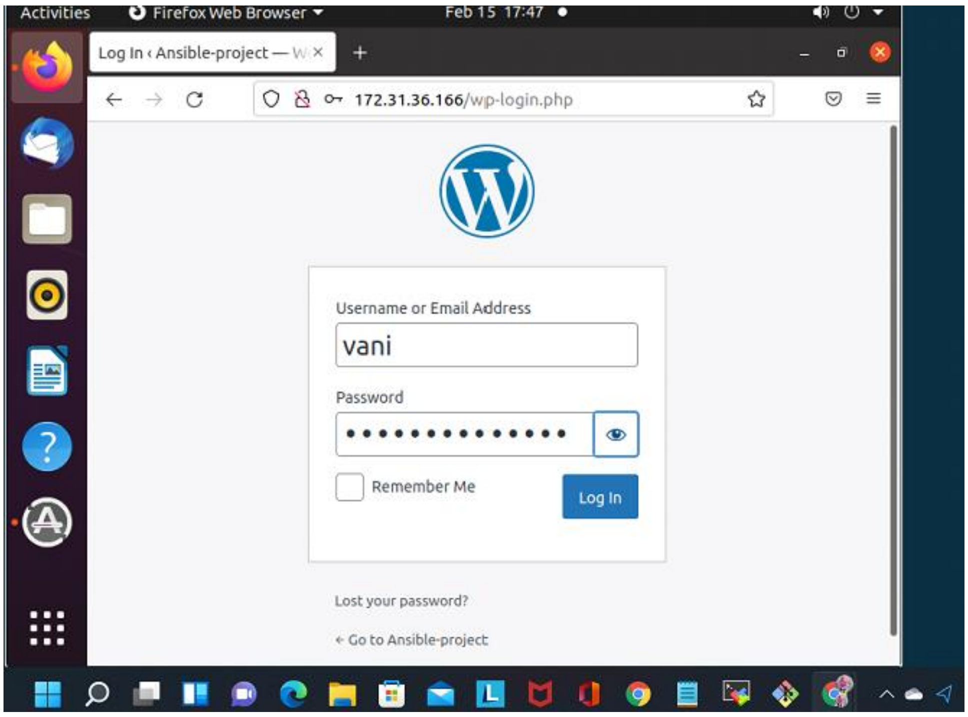
WordPress has been installed. Thank you, and enjoy!

Username

vani

Password

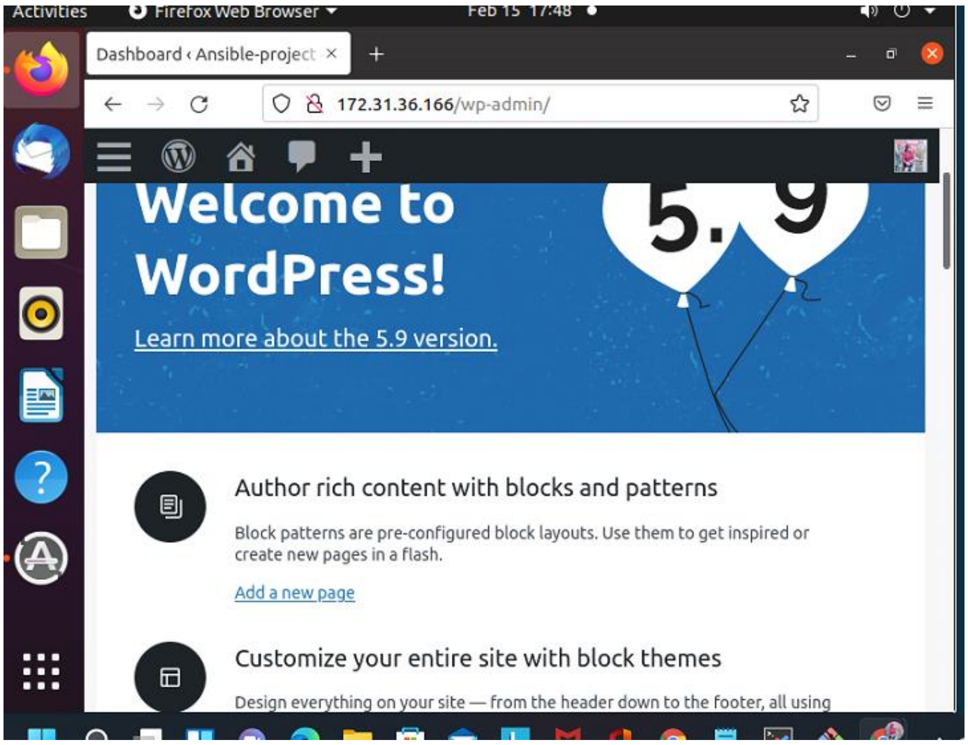
Your chosen password.



Username or Email Address: vani

Password: ••••••••••••••

Remember Me


Activities Firefox Web Browser Feb 15 17:48

Dashboard ‹ Ansible-project

172.31.36.166/wp-admin/

Welcome to WordPress!

Learn more about the 5.9 version.

Author rich content with blocks and patterns

Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.

Add a new page

Customize your entire site with block themes

Design everything on your site — from the header down to the footer, all using

Conclusion: -

I used Ansible to automate the process of installing and setting up a WordPress website with LAMP on an Ubuntu server.

Kalaivani.k

Vani212008@gmail.com