

Project assignment task description summary:

Through infrastructure automation my organization is adding more of the DevOps methodology and is specialized on WordPress projects. Through extending its infrastructure automation with configuration management a standard installation procedure for WordPress and its components is set up.

Components to be included:

PHP, Nginx/ Apache Web Server, MySQL, Wordpress

Important Steps:

Master connectivity is established with WordPress server through configuration management

The connectivity from the master to the slave machine is validated

A WordPress Installation script is established to install WordPress and its dependent components

The Scripts are executed which perform the installation of a complete WordPress environment

Through accessing a WordPress application using the public IP of the VM the installation is validated

Introduction	3
Environment Setup	3
Installing Ansible	3
Log in as Super User/ Root User	3
Apt install Ansible	3
Validating installation	4
Configure Ansible defaults	4
Verify Ansible defaults	4
Check installation files	4
SSH Connection Setup	5
Check Environment Hosts Information	5
Create Inventory File	5
Verify added Hosts	6
Setup SSH connection between master and host machines	7
Wordpress Environment Setup	8
Project specific Directory and inventory creation	8
Verify the SSH connection to host worker 1	9
Scripts creation and Execution	10
Playbook yaml file Execution 1	10
Apache Configuration file	12
Playbook yaml file Execution 2	13
Wordpress Configuration file	15
Playbook yaml file Execution 3	17
Wordpress Installation is verified	20

Introduction

For this assignment, the given Ubuntu Lab Environment is the basis workspace. There the Wordpress Environment is set up through Playbook Scripts with Ansible.

This is a reference installation guide used: <https://ubuntu.com/tutorials/install-and-configure-wordpress#1-overview>

Environment Setup

Installing Ansible

Log in as Super User/ Root User

Code:

sudo su -

Terminal:

```
labsuser@ip-172-31-27-100:~$ sudo su -
root@ip-172-31-27-100:~#
```

Apt install Ansible

Ansible is installed following this guide:

https://docs.ansible.com/ansible/latest/installation_guide/installation_distros.html#installing-ansible-on-ubuntu

Code:

\$ sudo apt update

\$ sudo apt install software-properties-common

\$ sudo add-apt-repository --yes --update ppa:ansible/ansible

\$ sudo apt install ansible

Terminal:

```
Selecting previously unselected package python3-winrm.
Preparing to unpack .../11-python3-winrm_0.3.0-2_all.deb ...
Unpacking python3-winrm (0.3.0-2) ...
Selecting previously unselected package sshpass.
Preparing to unpack .../12-sshpas_1.06-1_amd64.deb ...
Unpacking sshpass (1.06-1) ...
Setting up python3-ntlm-auth (1.1.0-1) ...
Setting up python3-resolvelib (0.5.4-2ppa~focal) ...
Setting up python3-kerberos (1.1.14-3.1build1) ...
Setting up sshpass (1.06-1) ...
Setting up python3-xmltodict (0.12.0-1) ...
Setting up python3-pyparsing (2.4.6-1) ...
Setting up python3-jmespath (0.9.4-2ubuntu1) ...
Setting up python3-requests-kerberos (0.12.0-2) ...
Setting up python3-requests-ntlm (1.1.0-1) ...
Setting up python3-packaging (20.3-1) ...
Setting up python3-winrm (0.3.0-2) ...
Setting up ansible-core (2.12.10-1ppa~focal) ...
Setting up ansible (5.10.0-1ppa~focal) ...
Processing triggers for man-db (2.9.1-1) ...
root@ip-172-31-27-100:~#
```

Validating installation

Code:

ansible - - version

Terminal:

```
root@ip-172-31-27-100:~# ansible --version
ansible [core 2.12.10]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.8.5 (default, Jan 27 2021, 15:41:15) [GCC 9.3.0]
  jinja version = 2.10.1
  libyaml = True
root@ip-172-31-27-100:~#
```

Configure Ansible defaults

Code:

```
echo "[defaults]
host_key_checking = False" > /etc/ansible/ansible.cfg
```

Terminal:

```
root@ip-172-31-27-100:~# echo "[defaults]
> host_key_checking = False" > /etc/ansible/ansible.cfg
root@ip-172-31-27-100:~#
```

Verify Ansible defaults

Code:

```
tail -n2 /etc/ansible/ansible.cfg
```

Terminal:

```
root@ip-172-31-27-100:~# tail -n2 /etc/ansible/ansible.cfg
[defaults]
host_key_checking = False
```

Check installation files

Code:

```
cd /etc/ansible/
ls
cat ansible.cfg
```

Terminal:

```
root@ip-172-31-27-100:~# cd /etc/ansible
root@ip-172-31-27-100:/etc/ansible# ls
ansible.cfg  hosts  roles
root@ip-172-31-27-100:/etc/ansible# cat ansible.cfg
[defaults]
host_key_checking = False
root@ip-172-31-27-100:/etc/ansible#
```

SSH Connection Setup

Check Environment Hosts Information

Lab Configuration Management Lab: Details

Details

Submission Details

Submission count: None

Due date: None

Username:

Status: Ready

Hostname	IP(public)	IP(private)	Port	Access
master	34.219.68.250	172.31.27.100	7681	http://34.219.68.250:7681
	8443	https://34.219.68.250:8443		
worker1	35.90.131.204	172.31.35.240	7681	http://35.90.131.204:7681
	8443	https://35.90.131.204:8443		
worker2	35.91.136.116	172.31.37.28	7681	http://35.91.136.116:7681
	8443	https://35.91.136.116:8443		

Worker 1: ip-172-31-35-240
(Worker 2: ip-172-31-37-28)

Create Inventory File

Worker 1 is called db
(Worker 2 is called web)

Code:

sudo vim /etc/ansible/hosts

Edit File to:

db ansible_host=ip-172-31-35-240
web ansible_host= ip-172-31-37-28

Terminal:

```
db ansible_host=ip-172-31-35-240
web ansible_host= ip-172-31-37-28
```

Verify added Hosts

Code:

ansible-inventory --list

Terminal:

```
root@ip-172-31-27-100:/etc/ansible# ansible-inventory --list
{
  "_meta": {
    "hostvars": {
      "db": {
        "ansible_host": "ip-172-31-35-240"
      },
      "web": {
        "ansible_host": "ip-172-31-37-28"
      }
    }
  },
  "all": {
    "children": [
      "ungrouped"
    ]
  },
  "ungrouped": {
    "hosts": [
      "db",
      "web"
    ]
  }
}
```

root@ip-172-31-27-100:/etc/ansible# █

Setup SSH connection between master and host machines

Code:

```
cd
ll ~/.ssh
ssh-keygen
(enter is pressed for prompts)
```

```
cat /root/.ssh/id_rsa.pub
```

Terminal:

```
root@ip-172-31-27-100:~# ll ~/.ssh
total 12
drwx----- 2 root root 4096 Mar 31 2021 ./
drwx----- 6 root root 4096 Oct 17 13:19 ../
-rw----- 1 root root 1104 Oct 17 12:29 authorized_keys
root@ip-172-31-27-100:~# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:13o23u9tov0ydwMwPMdJHEabet2dyt+f+bX8j+Ycd44 root@ip-172-31-27-100
The key's randomart image is:
+--[RSA 3072]--+
|                |
|      .+..+    |
|      ++..+    |
|  . + . +.+.   |
|    B 0  . .+  |
|    0 5  . .+  |
|  . @ .  . . .  |
|  . 0 .  0. +  |
|  .    .+B     |
|                |
+--[SHA256]--+
root@ip-172-31-27-100:~# cat /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDveiaA+T54jpvjf3dAsnqU0ZaKhWLKIvu+hInwvaK+qkNMZgt2o9dN/Bwd/F4CwWgW5ALXCjHIG1Vfo2D7y8o7w3DddsA641PQKvnp45+F
sMvCQ5qhexkzta8ljvtjTfvC2M1nIBkypLmeC3HCUeuFuSmTaulCsVCSRBr+tuelntqXqZqjNfFPod5bUfv8bAbO359T8UKO6VnRDfnUmNADYZHoB29qf3jYCgJNwlqrlgxW8L
hySOLPOHUNhCWRT0tVzeIHUGgrfXh8KG3P99C4XiGHdDtgujT4NdVT2d38rD96QBQ58QNI0qiobIEPZSdkiJdnVcTIkROUjDSQesrCkUjUfW6iYU9JsvU+fFEzzoV6+CdDMLaiEU8SJOKvX
ljsDqJ7ci5uUir6YQ9pMlIA2gEeDU6oT7WVxa9dBmqirMta0tNynPwtahCPezUHV0gWUgmJzjOIAxXg0rthODi0rYaPpjGO5laLpgZm71YsIbvkavsl0= root@ip-172-31-27-100
root@ip-172-31-27-100:~#
```

The Public Key is:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQGDveiaA+T54jpvjf3dAsnqU0ZaKhWLKIvu+hInwvaK+qkNMZgt2o9dN/Bwd/
F4CwWgW5ALXCjHIG1Vfo2D7y8o7w3DddsA641PQKvnp4S+F sMvCQ5qhexkzta8ljvtjTfvC2M1nIBkypLmeC3HCUeu
FuSmTaulCsVCSRBr+tuelntqXqZqjNfFPod5bUfv8bAbO359T8UKO6VnRDfnUmNADYZHoB29qf3jYCgJNwlqrlgxW8L
DKhe5PpeLeNhySOLPOHUNhCWRT0tVzeIHUGgrfXh8KG3P99C4XiGHdDtgujT4NdVT2dw38rD96QBQ58QNI0qiobIE
PZSdWijdnVcTIkROUjDSQesrCkUjUfW6iYU9JsvU+fFEzzoV6+CdDMLaiEU8SJOKvXljsDqJ7ci5uUUr6YQ9pMIIA2gE
eDU6oT7WVxa9dBmqirMta0tNynPwtahCPezUHV0gWUgmJzjOIAxXg0rthODi0rYaPpjGO5laLpgZm71YsIbvkavsl0=
root@ip-172-31-27-100
```

On Host Worker 1:

Code:

```
sudo su -
vim ~/.ssh/authorized_keys
```

In File:

Public Key is pasted

Wordpress Environment Setup

Project specific Directory and inventory creation

Project directory is created

Code:

mkdir cep

cd cep

Terminal:

```
root@ip-172-31-27-100:~# mkdir cep
root@ip-172-31-27-100:~# cd cep
```

Code:

vim wp-inventory

```
root@ip-172-31-27-100:~/cep# vim wp-inventory
```

In File:

wordpress ansible_host=ip-172-31-35-240

Code:

ansible-inventory -l -i wp-inventory

Terminal:

```
root@ip-172-31-27-100:~/cep# ansible-inventory --list -i wp-inventory
{
  "_meta": {
    "hostvars": {
      "wordpress": {
        "ansible_host": "ip-172-31-35-240"
      }
    }
  },
  "all": {
    "children": [
      "ungrouped"
    ]
  },
  "ungrouped": {
    "hosts": [
      "wordpress"
    ]
  }
}
```

```
root@ip-172-31-27-100:~/cep#
```

Verify the SSH connection to host worker 1

Code:

```
ansible wordpress -m ping -i wp-inventory
```

Terminal:

```
root@ip-172-31-27-100:~/cep# ansible wordpress -m ping -i wp-inventory
wordpress | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
root@ip-172-31-27-100:~/cep#
```

Scripts creation and Execution

Playbook yaml file Execution 1

Project playbook cep.yaml file is created:

Terminal:

vim cep.yaml

In File:

```
- name: "step1"
  hosts: wordpress
  vars:
  tasks:
    - name: "install mysql"
      apt:
        name: mysql-server
        update_cache: yes
    - name: "Install Dependencies"
      apt:
        name: "{{ item }}"
      loop:
        - apache2
        - ghostscript
        - libapache2-mod-php
        - php
        - php-bcmath
        - php-curl
        - php-imagick
        - php-intl
        - php-json
        - php-mbstring
        - php-mysql
        - php-xml
        - php-zip
        - python3-pymysql
```

Execute Playbook

Code:

```
ansible-playbook -i wp-inventory cep.yaml
```

Terminal:

```
root@ip-172-31-27-100:~/cep# ansible-playbook -i wp-inventory cep.yaml

PLAY [step1] *****
*

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

TASK [install mysql] *****
*
changed: [wordpress]

TASK [Install Dependencies] *****
*
changed: [wordpress] => (item=apache2)
ok: [wordpress] => (item=ghostscript)
changed: [wordpress] => (item=libapache2-mod-php)
changed: [wordpress] => (item=php)
changed: [wordpress] => (item=php-bcmath)
changed: [wordpress] => (item=php-curl)
changed: [wordpress] => (item=php-imagick)
changed: [wordpress] => (item=php-intl)
changed: [wordpress] => (item=php-json)
changed: [wordpress] => (item=php-mbstring)
changed: [wordpress] => (item=php-mysql)
changed: [wordpress] => (item=php-xml)
changed: [wordpress] => (item=php-zip)
changed: [wordpress] => (item=python3-pymysql)

PLAY RECAP *****
*
wordpress           : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

root@ip-172-31-27-100:~/cep#
```

Apache Configuration file

Code:

```
vim ~/wordpress-apache-config
```

In File:

```
<VirtualHost *:80>
    DocumentRoot /srv/www/wordpress
    <Directory /srv/www/wordpress>
        Options FollowSymLinks
        AllowOverride Limit Options FileInfo
        DirectoryIndex index.php
        Require all granted
    </Directory>
    <Directory /srv/www/wordpress/wp-content>
        Options FollowSymLinks
        Require all granted
    </Directory>
</VirtualHost>
```

Code:

```
mv ~/wordpress-apache-config .
```

```
ll
```

```
cat wordpress-apache-config
```

Terminal:

```
root@ip-172-31-27-100:~/cep# mv ~/wordpress-apache-config .
root@ip-172-31-27-100:~/cep# ll
total 20
drwxr-xr-x 2 root root 4096 Oct 17 22:47 ./
drwx----- 7 root root 4096 Oct 17 22:47 ../
-rw-r--r-- 1 root root  672 Oct 17 22:24 cep.yaml
-rw-r--r-- 1 root root  381 Oct 17 22:42 wordpress-apache-config
-rw-r--r-- 1 root root   40 Oct 17 21:37 wp-inventory
root@ip-172-31-27-100:~/cep# cat wordpress-apache-config
<VirtualHost *:80>
    DocumentRoot /srv/www/wordpress
    <Directory /srv/www/wordpress>
        Options FollowSymLinks
        AllowOverride Limit Options FileInfo
        DirectoryIndex index.php
        Require all granted
    </Directory>
    <Directory /srv/www/wordpress/wp-content>
        Options FollowSymLinks
        Require all granted
    </Directory>
</VirtualHost>
root@ip-172-31-27-100:~/cep#
```

Playbook yaml file Execution 2

Apache Configuration script processing has been added here:

In File:

```
- name: "step1"
  hosts: wordpress
  vars:
  tasks:
    - name: "install mysql"
      apt:
        name: mysql-server
        update_cache: yes
    - name: "Install Dependencies"
      apt:
        name: "{{ item }}"
      loop:
        - apache2
        - ghostscript
        - libapache2-mod-php
        - php
        - php-bcmath
        - php-curl
        - php-imagick
        - php-intl
        - php-json
        - php-mbstring
        - php-mysql
        - php-xml
        - php-zip
        - python3-pymysql

- name: "install wordpress"
  hosts: wordpress
  tasks:
    - name: "manage the installation directory"
      file:
        path: /srv/www
        state: directory
        recurse: yes
        owner: www-data
        group: www-data
    - name: "Download and extract the wordpress installation file"
      unarchive:
        src: https://wordpress.org/latest.tar.gz
        dest: /srv/www
        remote_src: yes

- name: "configure Apache for wordpress"
  hosts: wordpress
  tasks:
    - name: "define apache configuration file"
      copy:
        src: ~/cep/wordpress-apache-config
        dest: /etc/apache2/sites-available/wordpress.conf
    - name: "Enable apache modules"
      shell: "{{ item }}"
```

loop:

- a2ensite wordpress
 - a2enmod rewrite
 - a2dissite 000-default
 - service apache2 reload
-

Execute Playbook

Code:

ansible-playbook -i wp-inventory cep.yaml

Terminal:

```
root@ip-172-31-27-100:~/cep# ansible-playbook -i wp-inventory cep.yaml

PLAY [step1] *****

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

TASK [install mysql] *****
ok: [wordpress]

TASK [Install Dependencies] *****
ok: [wordpress] => (item=apache2)
ok: [wordpress] => (item=ghostscript)
ok: [wordpress] => (item=libapache2-mod-php)
ok: [wordpress] => (item=php)
ok: [wordpress] => (item=php-bcmath)
ok: [wordpress] => (item=php-curl)
ok: [wordpress] => (item=php-imagick)
ok: [wordpress] => (item=php-intl)
ok: [wordpress] => (item=php-json)
ok: [wordpress] => (item=php-mbstring)
ok: [wordpress] => (item=php-mysql)
ok: [wordpress] => (item=php-xml)
ok: [wordpress] => (item=php-zip)
ok: [wordpress] => (item=python3-pymysql)

PLAY [install wordpress] *****

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

TASK [manage the installation directory] *****
changed: [wordpress]

TASK [Download and extract the wordpress installation file] *****
changed: [wordpress]

PLAY [configure Apache for wordpress] *****

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

TASK [define apache configuration file] *****
changed: [wordpress]

TASK [Enable apache modules] *****
changed: [wordpress] => (item=a2ensite wordpress)
changed: [wordpress] => (item=a2enmod rewrite)
changed: [wordpress] => (item=a2dissite 000-default)
changed: [wordpress] => (item=service apache2 reload)

PLAY RECAP *****
wordpress          : ok=9    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

root@ip-172-31-27-100:~/cep#
```

Wordpress Configuration file

Code:

vim wp-config.php

In File:

```
<?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: *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH *
 * @link https://wordpress.org/documentation/article/editing-wp-config-php/ *
 * @package WordPress */
// ** Database settings - You can get this info from your web host ** // /** The name of the database for
WordPress */
define( 'DB_NAME', 'wordpress' );
/** Database username */
define( 'DB_USER', 'wordpress' ); /** Database password */
define( 'DB_PASSWORD', 'password' ); /** Database 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', '' );

/**#@+
 * 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', 'to`9jRmSZi+`r0;qJE=k_4}rn{ex]~)Q?]3X&!EV%B=+#yMFdql`SEL-zHJBW>+t');
define('SECURE_AUTH_KEY', 'gB8=-n!5eBf3Ua>Fhsuiu l1/F;Ow6nFo<.-;bc;lqMDEz1etC}9;yg.5dbJ+[e;');
define('LOGGED_IN_KEY', 'S<FvJlpIYJiS*u)Gx31:p+buO[R-=S4l_)#vo)M(HT8`Q8PVtru$-csYpal-a{jI');
define('NONCE_KEY', 'G~ldZ%_DV#`(#- ROclQIH+&o$(YgPcZ<GE:r)6Xim^+l%t1ln8`#ehAfV9>XR3C');
define('AUTH_SALT', '1wgW`5LjVQMI2BZeBTP K_)!Pa8yVfRwCliz*=-;V:iG:Sgt}B-20]pY%GyWn+x');
define('SECURE_AUTH_SALT', ':uKK2-W2KXvghvlgszdEq,=Q2y4QSYI7Lx{3WylNaMya S?mc0l6pNXH,k!
x?+vS');
define('LOGGED_IN_SALT', ',.L@;VjZJ;!gohAV/GRp*4]k&U+qs(leFtHE`gd%_osl){SrlD0c)M@IMg/C2]L');
define('NONCE_SALT', 'dR7s4b8N9<hyhG;z_>FWt1Ah)qlgKIXH?l`C -l7(H2lAkGT&b4OhKINE)n$G$G{u');
/**#@-*/

/**
 * 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 documentation. *  
* @link https://wordpress.org/documentation/article/debugging-in-wordpress/ */
```

```
define( 'WP_DEBUG', false );
```

```
/* Add any custom values between this line and the "stop editing" line. */
```

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

```
/** Absolute path to the WordPress directory. */
```

```
if ( ! defined( 'ABSPATH' ) ) {  
    define( 'ABSPATH', __DIR__ . '/' );  
}
```

```
/** Sets up WordPress vars and included files. */
```

```
require_once ABSPATH . 'wp-settings.php';
```

```
-----
```


Playbook yaml file Execution 3

Wordpress Configuration script processing has been added here:

Terminal:

vim cep.yaml

In File:

```
- name: "step1"
  hosts: wordpress
  vars:
  tasks:
    - name: "install mysql"
      apt:
        name: mysql-server
        update_cache: yes
    - name: "Install Dependencies"
      apt:
        name: "{{ item }}"
      loop:
        - apache2
        - ghostscript
        - libapache2-mod-php
        - php
        - php-bcmath
        - php-curl
        - php-imagick
        - php-intl
        - php-json
        - php-mbstring
        - php-mysql
        - php-xml
        - php-zip
        - python3-pymysql

- name: "install wordpress"
  hosts: wordpress
  tasks:
    - name: "manage the installation directory"
      file:
        path: /srv/www
        state: directory
        recurse: yes
        owner: www-data
        group: www-data
    - name: "Download and extract the wordpress installation file"
      unarchive:
        src: https://wordpress.org/latest.tar.gz
        dest: /srv/www
        remote_src: yes

- name: "configure Apache for wordpress"
  hosts: wordpress
  tasks:
    - name: "define apache configuration file"
      copy:
        src: ~/cep/wordpress-apache-config
```

```
    dest: /etc/apache2/sites-available/wordpress.conf
- name: "Enable apache modules"
  shell: "{{ item }}"
  loop:
    - a2ensite wordpress
    - a2enmod rewrite
    - a2dissite 000-default
    - service apache2 reload
- name: "create DB and manage DB user and add DB details to WP config"
  hosts: wordpress
  tasks:
    - name: "create required DB"
      mysql_db:
        name: wordpress
        state: present
        login_unix_socket: /var/run/mysqld/mysqld.sock
    - name: "Manage DB user and permissions"
      mysql_user:
        name: wordpress
        password: "password"
        priv: '*.*:ALL'
        state: present
        login_unix_socket: /var/run/mysqld/mysqld.sock
- name: "wordpress configuration file"
  hosts: wordpress
  tasks:
    - name: "copy WP config file"
      copy:
        src: ~/cep/wp-config.php
        dest: /srv/www/wordpress/wp-config.php
```

Execute Playbook

Code:

ansible-playbook -i wp-inventory cep.yaml

Terminal:

```
root@ip-172-31-27-100:~/cep# vim cep.yaml
root@ip-172-31-27-100:~/cep# ansible-playbook -i wp-inventory cep.yaml

PLAY [step1] *****

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

TASK [install mysql] *****
ok: [wordpress]

TASK [Install Dependencies] *****
ok: [wordpress] => (item=apache2)
ok: [wordpress] => (item=ghostscript)
ok: [wordpress] => (item=libapache2-mod-php)
ok: [wordpress] => (item=php)
ok: [wordpress] => (item=php-bcmath)
ok: [wordpress] => (item=php-curl)
ok: [wordpress] => (item=php-imagick)
ok: [wordpress] => (item=php-intl)
ok: [wordpress] => (item=php-json)
ok: [wordpress] => (item=php-mbstring)
ok: [wordpress] => (item=php-mysql)
ok: [wordpress] => (item=php-xml)
ok: [wordpress] => (item=php-zip)
ok: [wordpress] => (item=python3-pymysql)

PLAY [install wordpress] *****

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

TASK [manage the installation directory] *****
changed: [wordpress]

TASK [Download and extract the wordpress installation file] *****
changed: [wordpress]

PLAY [configure Apache for wordpress] *****

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

TASK [define apache configuration file] *****
ok: [wordpress]

TASK [Enable apache modules] *****
changed: [wordpress] => (item=a2ensite wordpress)
changed: [wordpress] => (item=a2enmod rewrite)
changed: [wordpress] => (item=a2dissite 000-default)
changed: [wordpress] => (item=service apache2 reload)

PLAY [create DB and manage DB user and add DB details to WP config] *****

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

TASK [create required DB] *****
changed: [wordpress]

TASK [Manage DB user and permissions] *****
changed: [wordpress]

PLAY [wordpress configuration file] *****

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

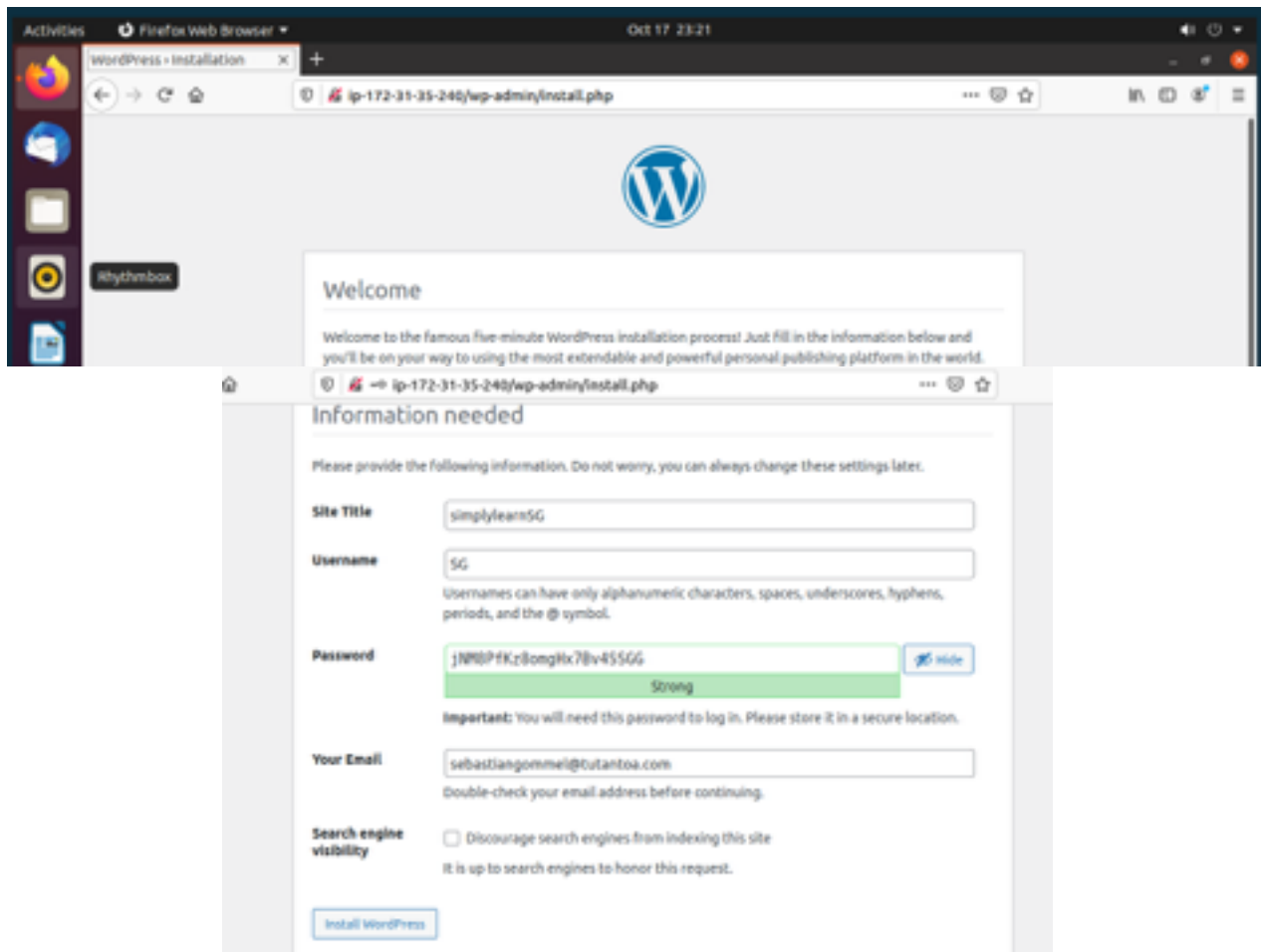
TASK [copy WP config file] *****
changed: [wordpress]

PLAY RECAP *****
wordpress : ok=14 changed=6 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

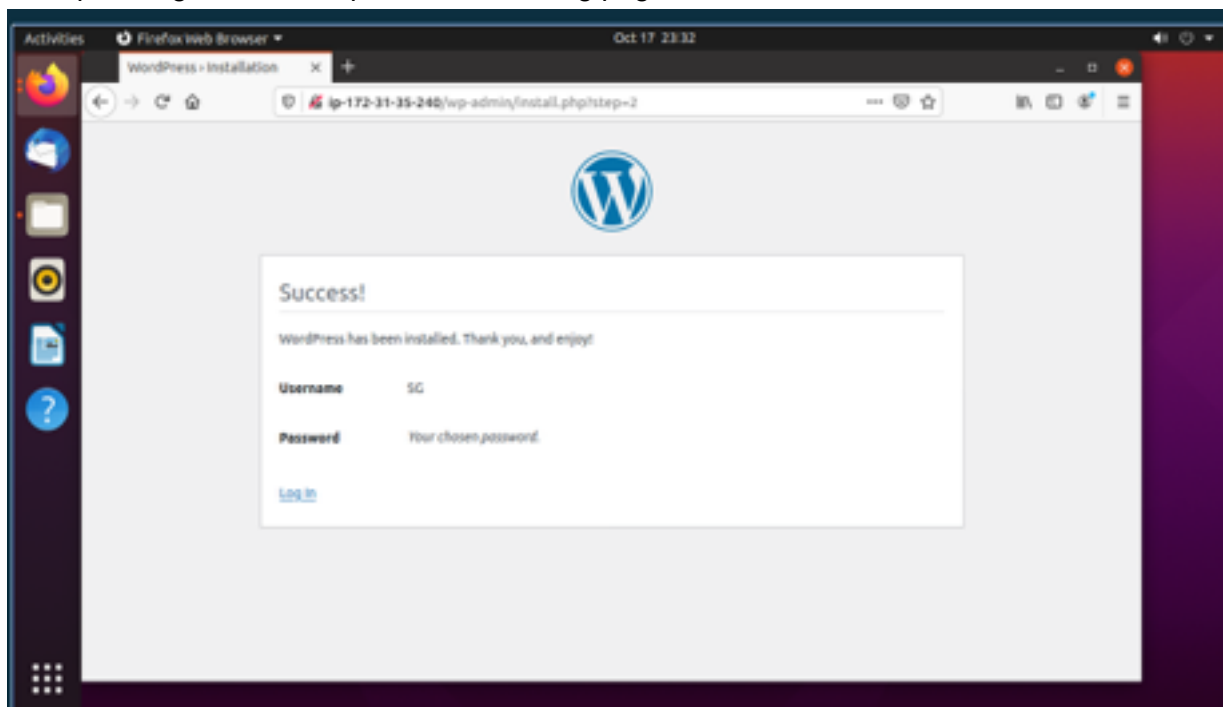
root@ip-172-31-27-100:~/cep#
```

Wordpress Installation is verified

The lab is opened; the master desktop is opened and on the following URL is browsed in Firefox
<http://ip-172-31-35-240>



After pressing Install Wordpress the following page is shown:



The following address allows to login to Wordpress
ip-172-31-35-240/wp-login.php

