Bahman Sheikh

Deployment of WordPress Environment

DESCRIPTION

You are a DevOps engineer at XYZ Ltd. Your company is working mostly on WordPress projects. A lot of development hours are lost to perform WordPress setup with all dependencies like PHP, MySQL, etc. The Company wants to automate it with the help of a configuration management tool so that they can follow a standard installation procedure for WordPress and its components whenever a new requirement or client comes in. The below mentioned components should be included:

- PHP
- Nginx/Apache Web Server
- MySQL
- WordPress

Steps to Perform:

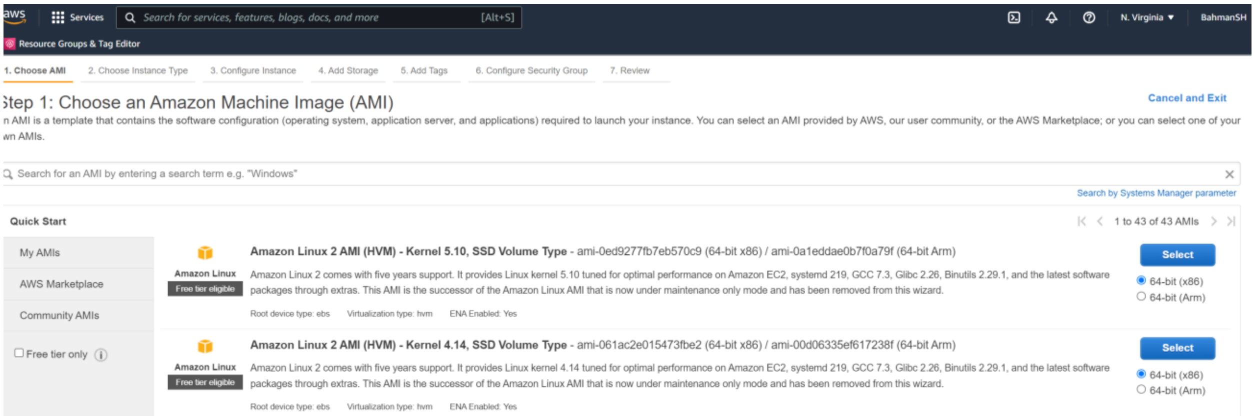
1. Establish configuration management master connectivity with WordPress server
2. Validate connectivity from master to slave machine
3. Prepare IaaC scripts to install WordPress and its dependent components
4. Execute scripts to perform installation of complete WordPress environment
5. Validate installation using the public IP of VM by accessing WordPress application

Codes also available in my GitHub here:

https://github.com/SheikhBahman/Caltech_DevOps/tree/main/PG_DO_Configuration_Management_with_Chef_Puppet_and_Ansible/bahmanProjectAnsibleWordPress/Attempt_AWS_YUM

Project Steps, codes and screenshots:

1. Create two AWS instances one as the master and the other for the WordPress server:



Step 1: Choose an Amazon Machine Image (AMI)

n AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your wn AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-0ed9277fb7eb570c9 (64-bit x86) / ami-0a1eddae0b7f0a79f (64-bit Arm)

Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

● 64-bit (x86)
○ 64-bit (Arm)

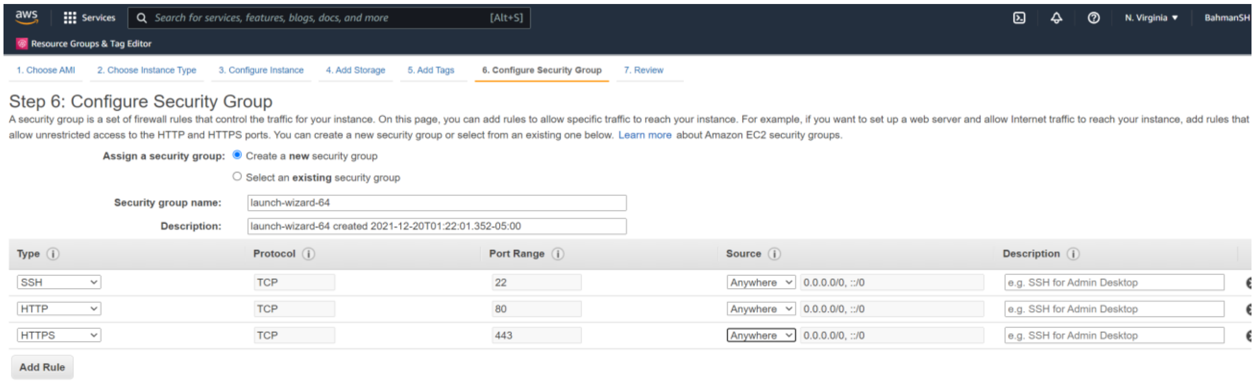
Amazon Linux 2 AMI (HVM) - Kernel 4.14, SSD Volume Type - ami-061ac2e015473fbe2 (64-bit x86) / ami-00d06335ef617238f (64-bit Arm)

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

● 64-bit (x86)
○ 64-bit (Arm)

2. Make sure SSH and http traffic are allowed:



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

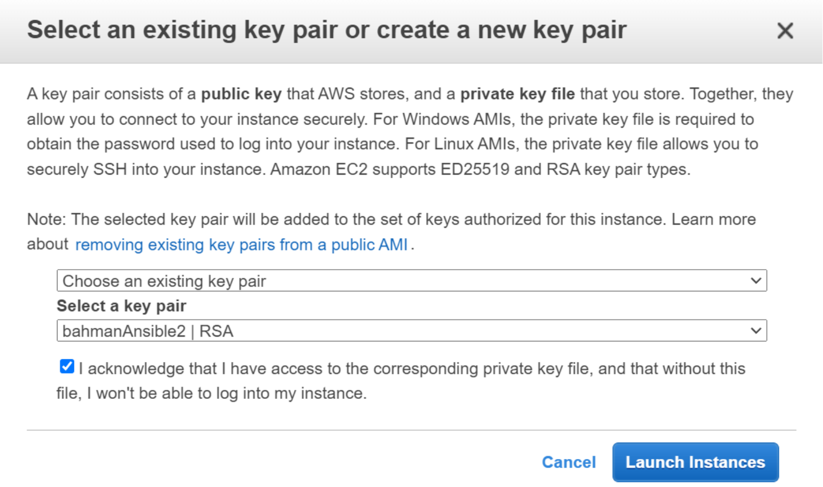
Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: launch-wizard-64

Description: launch-wizard-64 created 2021-12-20T01:22:01.352-05:00

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

3. key-pair for Ansible connection: bahmanAnsible2.pem



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

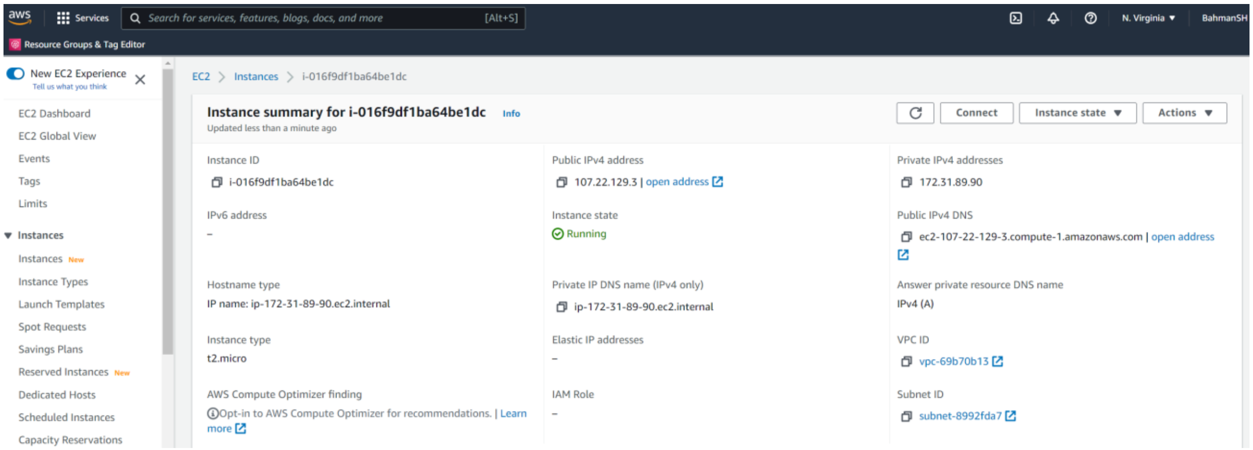
Choose an existing key pair

Select a key pair

bahmanAnsible2 | RSA

☑ I acknowledge that I have access to the corresponding private key file, and that without this file, I won't be able to log into my instance.

Master node:



Instance summary for i-016f9df1ba64be1dc
Updated less than a minute ago

Instance ID
i-016f9df1ba64be1dc

Public IPv4 address
107.22.129.3 | open address

Private IPv4 addresses
172.31.89.90

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-107-22-129-3.compute-1.amazonaws.com | open address

Hostname type
IP name: ip-172-31-89-90.ec2.internal

Private IP DNS name (IPv4 only)
ip-172-31-89-90.ec2.internal

Answer private resource DNS name
IPv4 (A)

Instance type
t2.micro

Elastic IP addresses
–

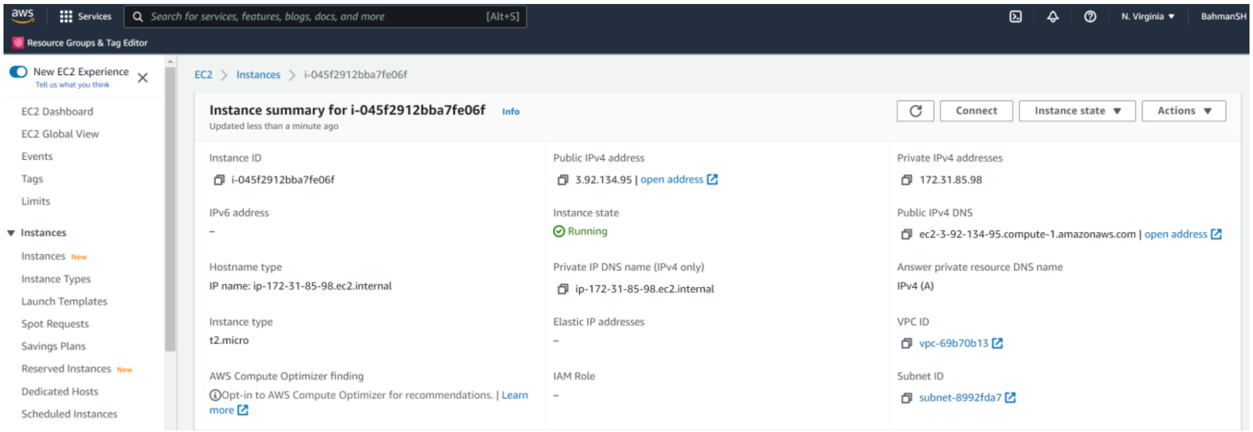
VPC ID
vpc-69b70b13

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role
–

Subnet ID
subnet-8992fda7

Worker node or the WordPress server:



Instance summary for i-045f2912bba7fe06f

Updated less than a minute ago

Instance ID: i-045f2912bba7fe06f

Public IPv4 address: 3.92.134.95 | open address

Private IPv4 addresses: 172.31.85.98

IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-3-92-134-95.compute-1.amazonaws.com | open address

Hostname type: IP name: ip-172-31-85-98.ec2.internal

Private IP DNS name (IPv4 only): ip-172-31-85-98.ec2.internal

Answer private resource DNS name: IPv4 (A)

Instance type: t2.micro

Elastic IP addresses: –

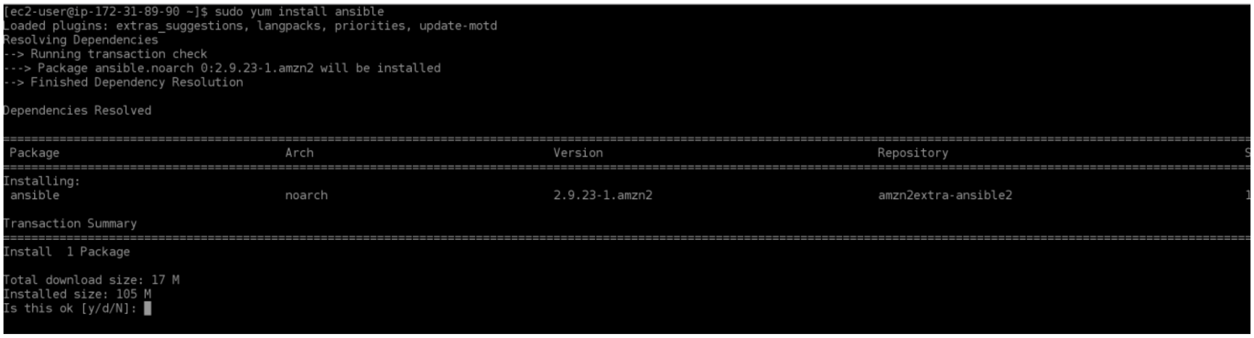
VPC ID: vpc-69b70b13

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role: –

Subnet ID: subnet-8992fda7

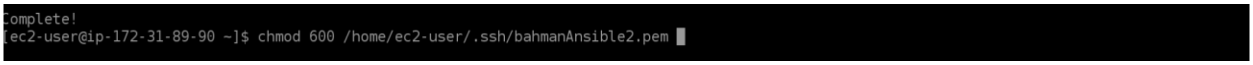
4. First install Ansible on the Master node:



```
[ec2-user@ip-172-31-89-90 ~]$ sudo yum install ansible
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package ansible.noarch 0:2.9.23-1.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================================================
 Package                              Arch                            Version                                    Repository                    S
================================================================================================================================================
Installing:
 ansible                              noarch                          2.9.23-1.amzn2                             amzn2extra-ansible2           1

Transaction Summary
================================================================================================================================================
Install  1 Package

Total download size: 17 M
Installed size: 105 M
Is this ok [y/d/N]:
```

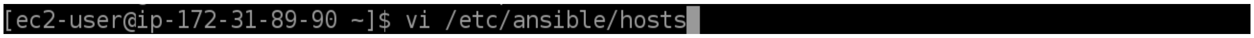
5. Copy the pair key file to the master and configure the Ansible hosts:

Key file and permission

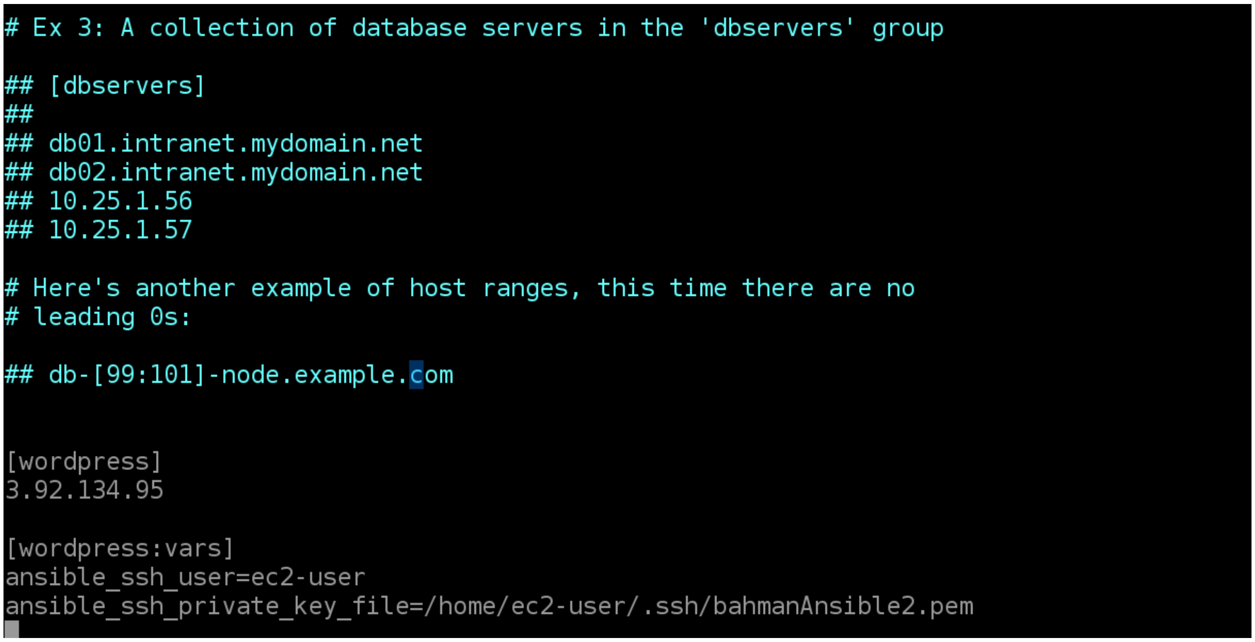


```
Complete!
[ec2-user@ip-172-31-89-90 ~]$ chmod 600 /home/ec2-user/.ssh/bahmanAnsible2.pem
```

6. Ansible hosts and SSH user credentials:

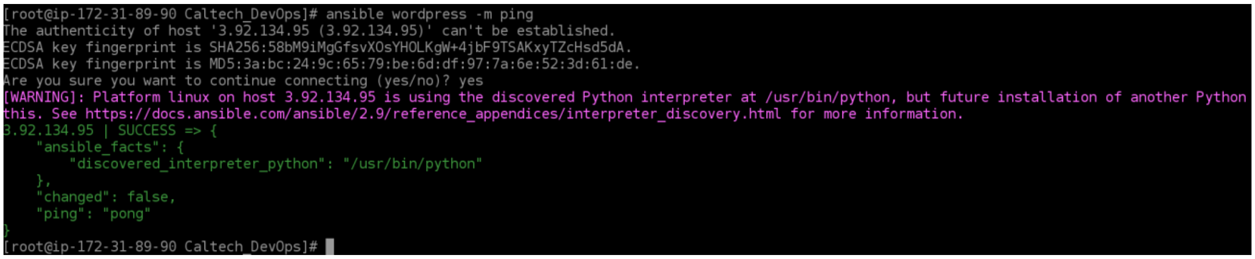


```
[ec2-user@ip-172-31-89-90 ~]$ vi /etc/ansible/hosts
```



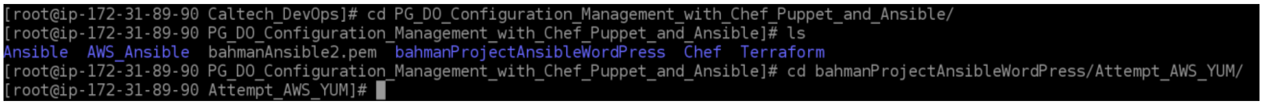
```
# Ex 3: A collection of database servers in the 'dbservers' group

## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# Here's another example of host ranges, this time there are no
# leading 0s:

## db-[99:101]-node.example.com


[wordpress]
3.92.134.95

[wordpress:vars]
ansible_ssh_user=ec2-user
ansible_ssh_private_key_file=/home/ec2-user/.ssh/bahmanAnsible2.pem
```

7. Ping to check the connectivity with the node machine:



```
[root@ip-172-31-89-90 Caltech_DevOps]# ansible wordpress -m ping
The authenticity of host '3.92.134.95 (3.92.134.95)' can't be established.
ECDSA key fingerprint is SHA256:58bM9iMgGfsvXOsYHOLKgW+4jbF9TSAKxyTZcHsd5dA.
ECDSA key fingerprint is MD5:3a:bc:24:9c:65:79:be:6d:df:97:7a:6e:52:3d:61:de.
Are you sure you want to continue connecting (yes/no)? yes
[WARNING]: Platform linux on host 3.92.134.95 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python
this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
3.92.134.95 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[root@ip-172-31-89-90 Caltech_DevOps]#
```

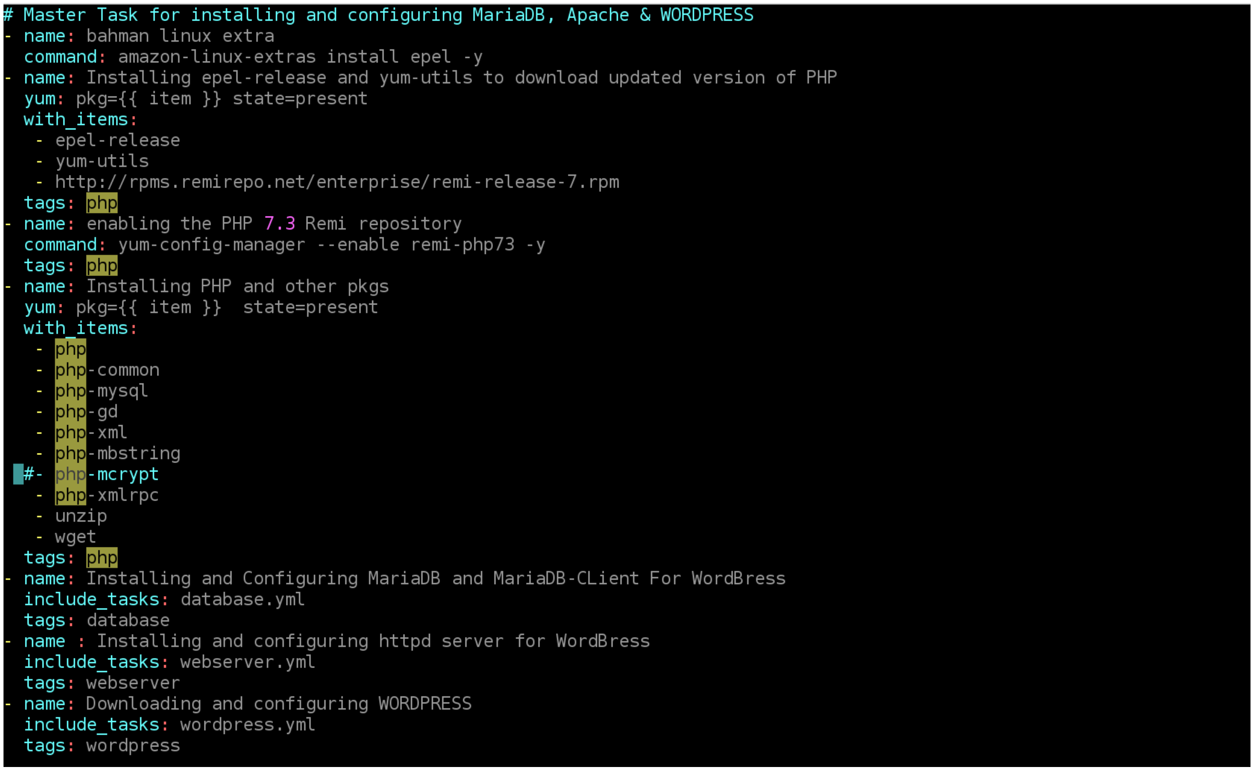
8. Create the project folder:



```
[root@ip-172-31-89-90 Caltech_DevOps]# cd PG_DO_Configuration_Management_with_Chef_Puppet_and_Ansible/
[root@ip-172-31-89-90 PG_DO_Configuration_Management_with_Chef_Puppet_and_Ansible]# ls
Ansible  AWS_Ansible  bahmanAnsible2.pem  bahmanProjectAnsibleWordPress  Chef  Terraform
[root@ip-172-31-89-90 PG_DO_Configuration_Management_with_Chef_Puppet_and_Ansible]# cd bahmanProjectAnsibleWordPress/Attempt_AWS_YUM/
[root@ip-172-31-89-90 Attempt_AWS_YUM]#
```

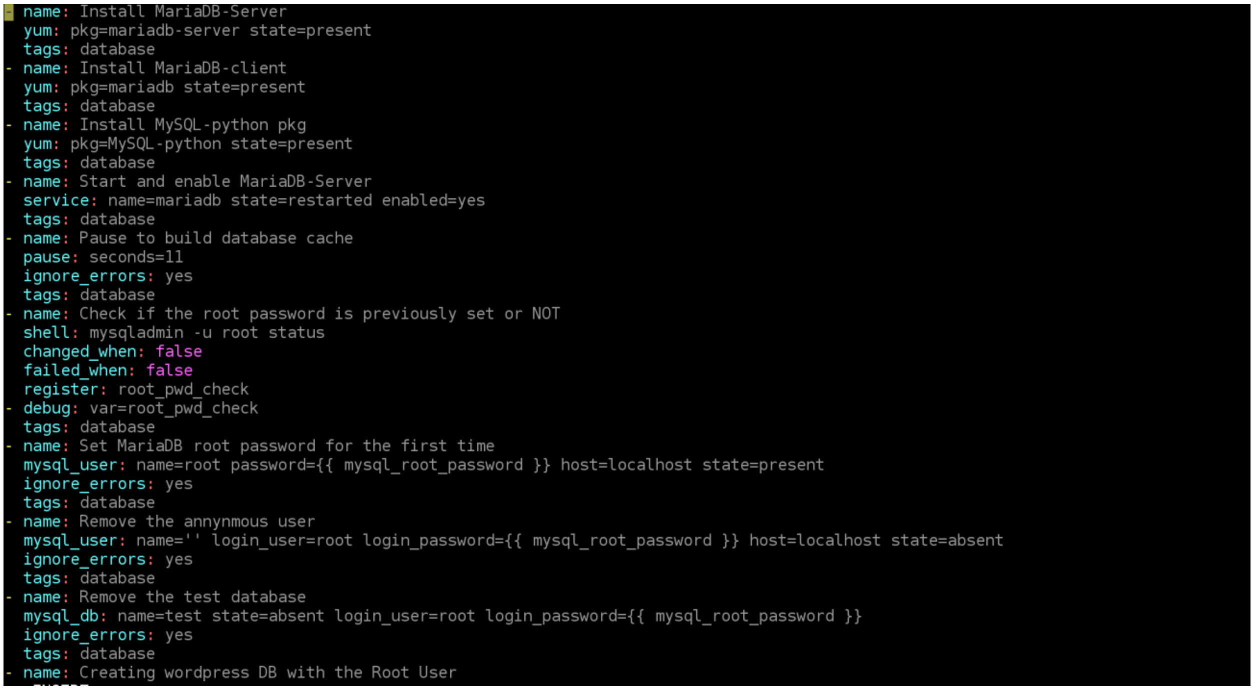
9. Crease Ansible task files as:

1. main.yml

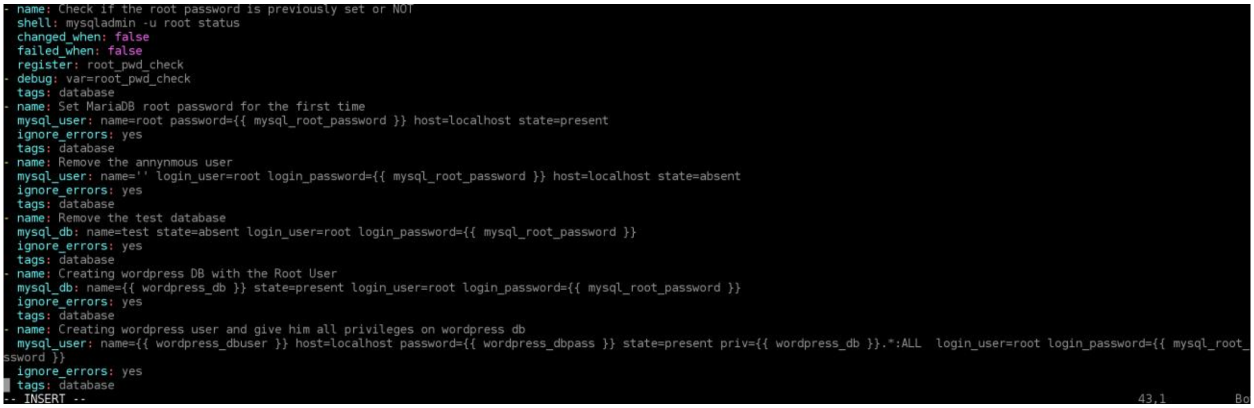


```
# Master Task for installing and configuring MariaDB, Apache & WORDPRESS
- name: bahman linux extra
  command: amazon-linux-extras install epel -y
- name: Installing epel-release and yum-utils to download updated version of PHP
  yum: pkg={{ item }} state=present
  with_items:
   - epel-release
   - yum-utils
   - http://rpms.remirepo.net/enterprise/remi-release-7.rpm
  tags: php
- name: enabling the PHP 7.3 Remi repository
  command: yum-config-manager --enable remi-php73 -y
  tags: php
- name: Installing PHP and other pkgs
  yum: pkg={{ item }}  state=present
  with_items:
   - php
   - php-common
   - php-mysql
   - php-gd
   - php-xml
   - php-mbstring
  #- php-mcrypt
   - php-xmlrpc
   - unzip
   - wget
  tags: php
- name: Installing and Configuring MariaDB and MariaDB-CLient For WordBress
  include_tasks: database.yml
  tags: database
- name : Installing and configuring httpd server for WordBress
  include_tasks: webserver.yml
  tags: webserver
- name: Downloading and configuring WORDPRESS
  include_tasks: wordpress.yml
  tags: wordpress
```

2. database.yml

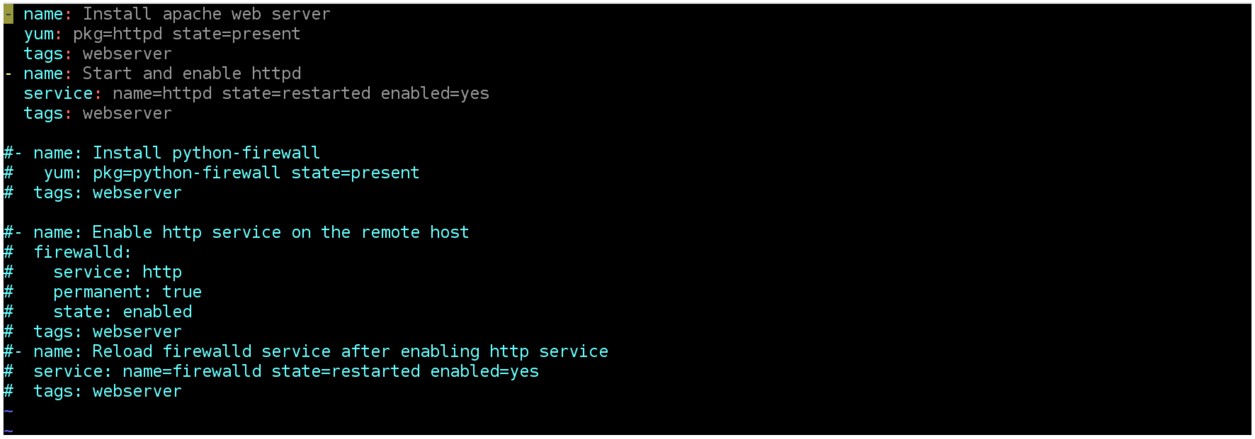


```
- name: Install MariaDB-Server
  yum: pkg=mariadb-server state=present
  tags: database
- name: Install MariaDB-client
  yum: pkg=mariadb state=present
  tags: database
- name: Install MySQL-python pkg
  yum: pkg=MySQL-python state=present
  tags: database
- name: Start and enable MariaDB-Server
  service: name=mariadb state=restarted enabled=yes
  tags: database
- name: Pause to build database cache
  pause: seconds=11
  ignore_errors: yes
  tags: database
- name: Check if the root password is previously set or NOT
  shell: mysqladmin -u root status
  changed_when: false
  failed_when: false
  register: root_pwd_check
- debug: var=root_pwd_check
  tags: database
- name: Set MariaDB root password for the first time
  mysql_user: name=root password={{ mysql_root_password }} host=localhost state=present
  ignore_errors: yes
  tags: database
- name: Remove the annynmous user
  mysql_user: name='' login_user=root login_password={{ mysql_root_password }} host=localhost state=absent
  ignore_errors: yes
  tags: database
- name: Remove the test database
  mysql_db: name=test state=absent login_user=root login_password={{ mysql_root_password }}
  ignore_errors: yes
  tags: database
- name: Creating wordpress DB with the Root User
```



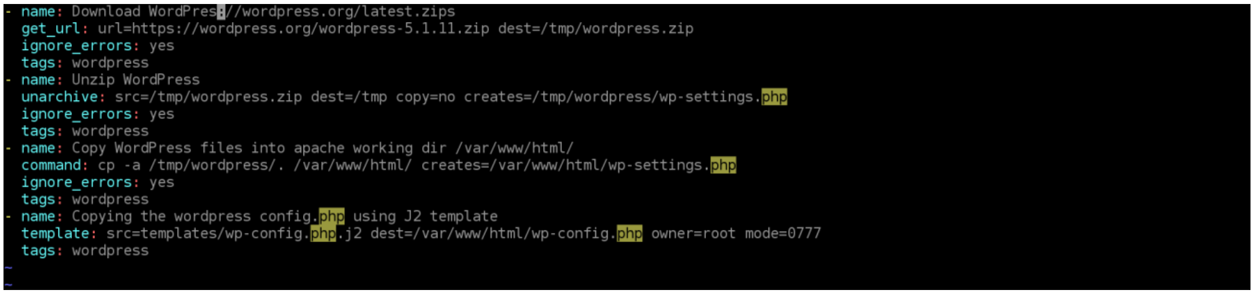
```
- name: Check if the root password is previously set or NOT
  shell: mysqladmin -u root status
  changed_when: false
  failed_when: false
  register: root_pwd_check
- debug: var=root_pwd_check
  tags: database
- name: Set MariaDB root password for the first time
  mysql_user: name=root password={{ mysql_root_password }} host=localhost state=present
  ignore_errors: yes
  tags: database
- name: Remove the annynmous user
  mysql_user: name='' login_user=root login_password={{ mysql_root_password }} host=localhost state=absent
  ignore_errors: yes
  tags: database
- name: Remove the test database
  mysql_db: name=test state=absent login_user=root login_password={{ mysql_root_password }}
  ignore_errors: yes
  tags: database
- name: Creating wordpress DB with the Root User
  mysql_db: name={{ wordpress_db }} state=present login_user=root login_password={{ mysql_root_password }}
  ignore_errors: yes
  tags: database
- name: Creating wordpress user and give him all privileges on wordpress db
  mysql_user: name={{ wordpress_dbuser }} host=localhost password={{ wordpress_dbpass }} state=present priv={{ wordpress_db }}.*:ALL  login_user=root login_password={{ mysql_root_
ssword }}
  ignore_errors: yes
  tags: database
-- INSERT --                                                                                                                                                   43,1          Bo
```

3. webserver.yml



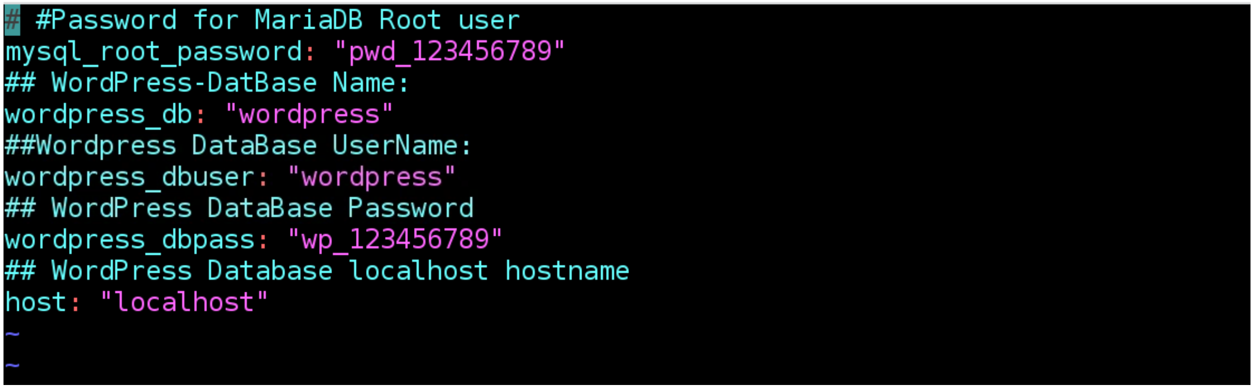
```
- name: Install apache web server
  yum: pkg=httpd state=present
  tags: webserver
- name: Start and enable httpd
  service: name=httpd state=restarted enabled=yes
  tags: webserver

#- name: Install python-firewall
#   yum: pkg=python-firewall state=present
#  tags: webserver

#- name: Enable http service on the remote host
#  firewalld:
#    service: http
#    permanent: true
#    state: enabled
#  tags: webserver
#- name: Reload firewalld service after enabling http service
#  service: name=firewalld state=restarted enabled=yes
#  tags: webserver
~
~
```

4. werdpress.yml

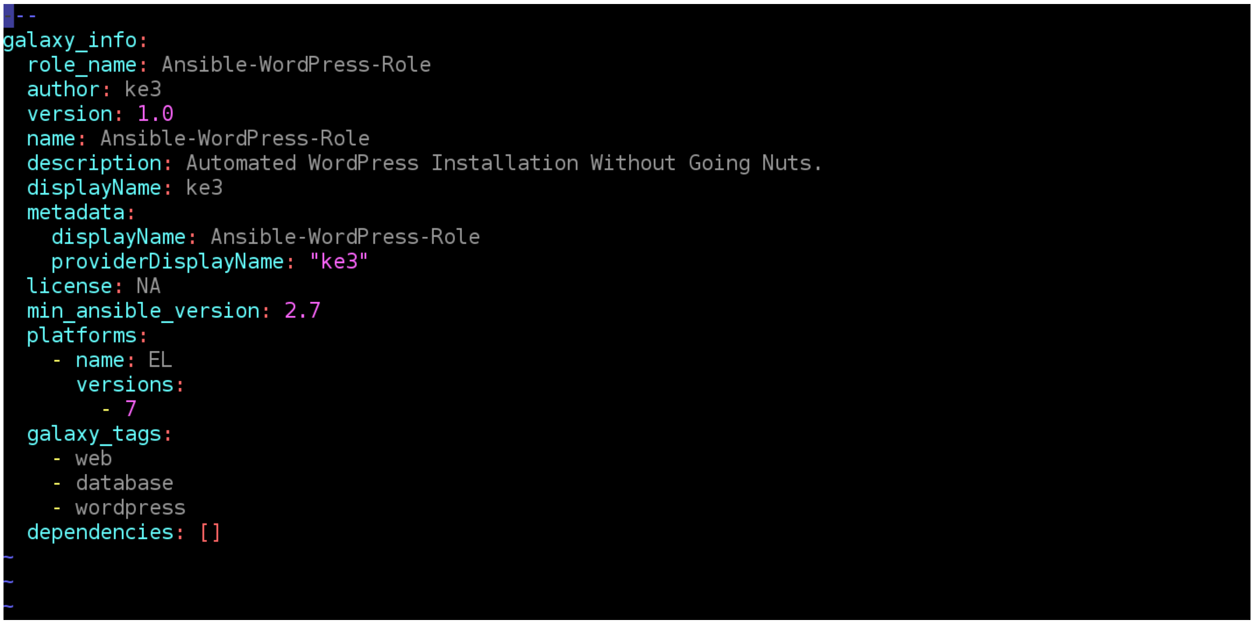


```
- name: Download WordPres://wordpress.org/latest.zips
  get_url: url=https://wordpress.org/wordpress-5.1.11.zip dest=/tmp/wordpress.zip
  ignore_errors: yes
  tags: wordpress
- name: Unzip WordPress
  unarchive: src=/tmp/wordpress.zip dest=/tmp copy=no creates=/tmp/wordpress/wp-settings.php
  ignore_errors: yes
  tags: wordpress
- name: Copy WordPress files into apache working dir /var/www/html/
  command: cp -a /tmp/wordpress/. /var/www/html/ creates=/var/www/html/wp-settings.php
  ignore_errors: yes
  tags: wordpress
- name: Copying the wordpress config.php using J2 template
  template: src=templates/wp-config.php.j2 dest=/var/www/html/wp-config.php owner=root mode=0777
  tags: wordpress
~
~
```

10. Ansible variables:



```
# #Password for MariaDB Root user
mysql_root_password: "pwd_123456789"
## WordPress-DatBase Name:
wordpress_db: "wordpress"
##Wordpress DataBase UserName:
wordpress_dbuser: "wordpress"
## WordPress DataBase Password
wordpress_dbpass: "wp_123456789"
## WordPress Database localhost hostname
host: "localhost"
~
~
```



```yaml
---
galaxy_info:
  role_name: Ansible-WordPress-Role
  author: ke3
  version: 1.0
  name: Ansible-WordPress-Role
  description: Automated WordPress Installation Without Going Nuts.
  displayName: ke3
  metadata:
    displayName: Ansible-WordPress-Role
    providerDisplayName: "ke3"
  license: NA
  min_ansible_version: 2.7
  platforms:
    - name: EL
      versions:
        - 7
  galaxy_tags:
    - web
    - database
    - wordpress
  dependencies: []
~
~
~
```

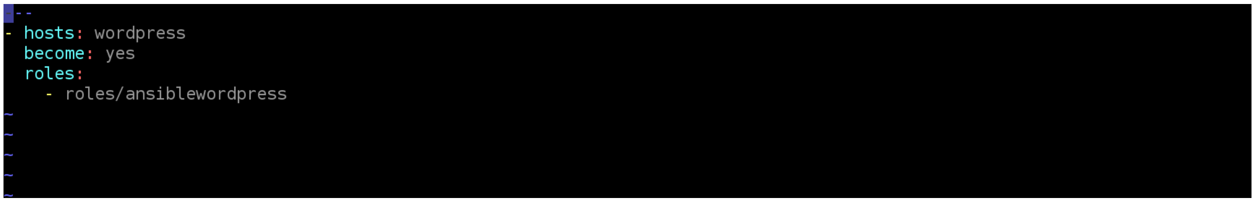
11. WordPress config file:

```php
<?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', '{{ wordpress_db }}');
```

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


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


/** 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', '');


/**#@+
 * 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',         'put your unique phrase here');
define('SECURE_AUTH_KEY',  'put your unique phrase here');
define('LOGGED_IN_KEY',    'put your unique phrase here');
define('NONCE_KEY',        'put your unique phrase here');
define('AUTH_SALT',        'put your unique phrase here');
define('SECURE_AUTH_SALT', 'put your unique phrase here');
define('LOGGED_IN_SALT',   'put your unique phrase here');
define('NONCE_SALT',       'put your unique phrase here');


/**#@-*/
```

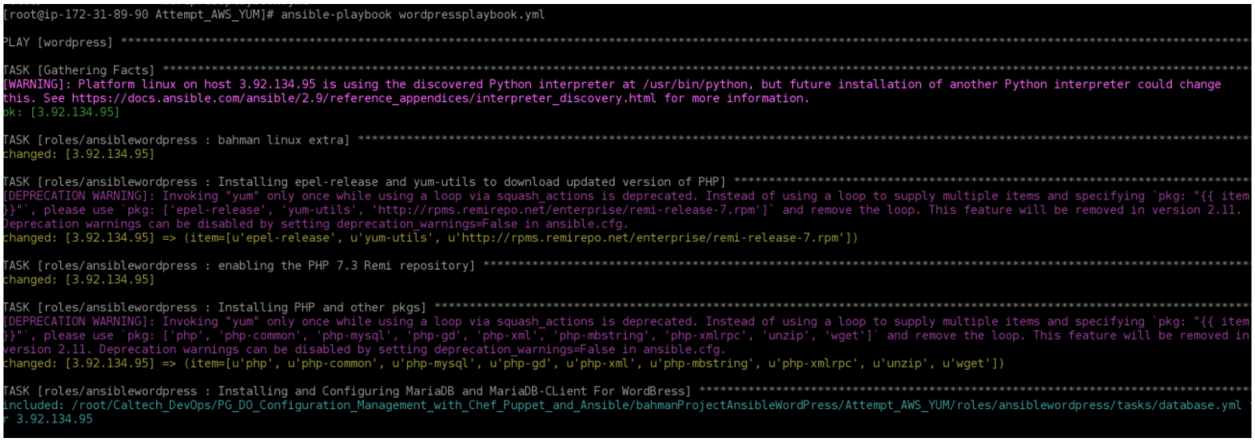
```
/**
 * WordPress Database Table prefix.
 *
 * You can have multiple installations in one database if you give each
 * a unique prefix. Only numbers, letters, and underscores please!
 */
$table_prefix  = 'wp_';


/**
 * 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 blogging. */


/** 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');
```

12. Main Ansible playbook

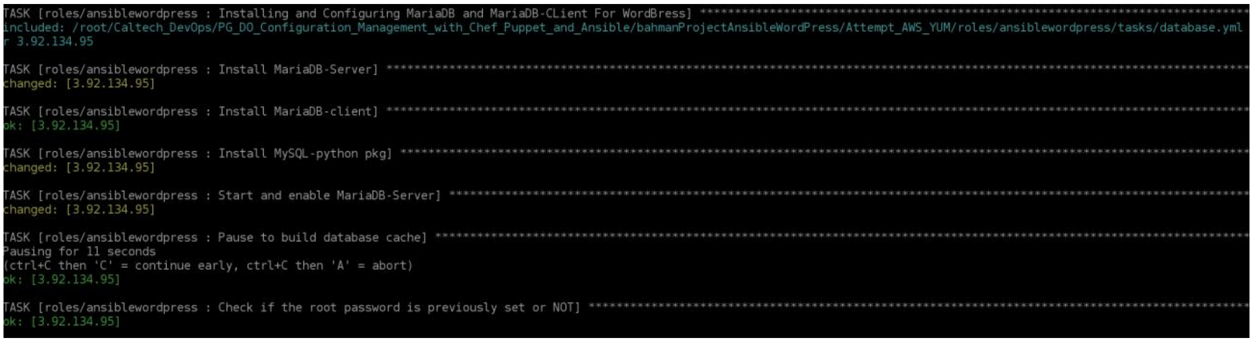


```
---
- hosts: wordpress
  become: yes
  roles:
    - roles/ansiblewordpress
~
~
~
~
~
```

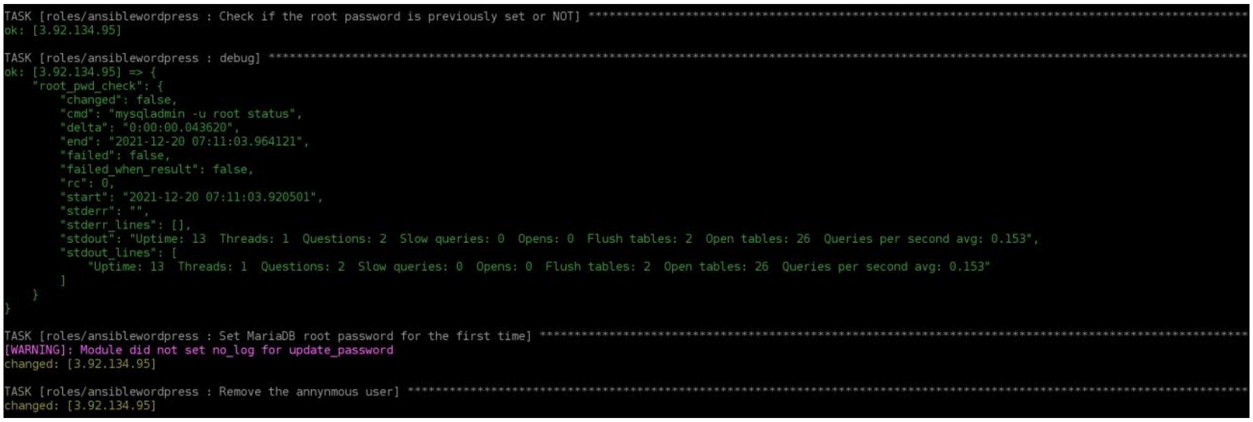
13. Run



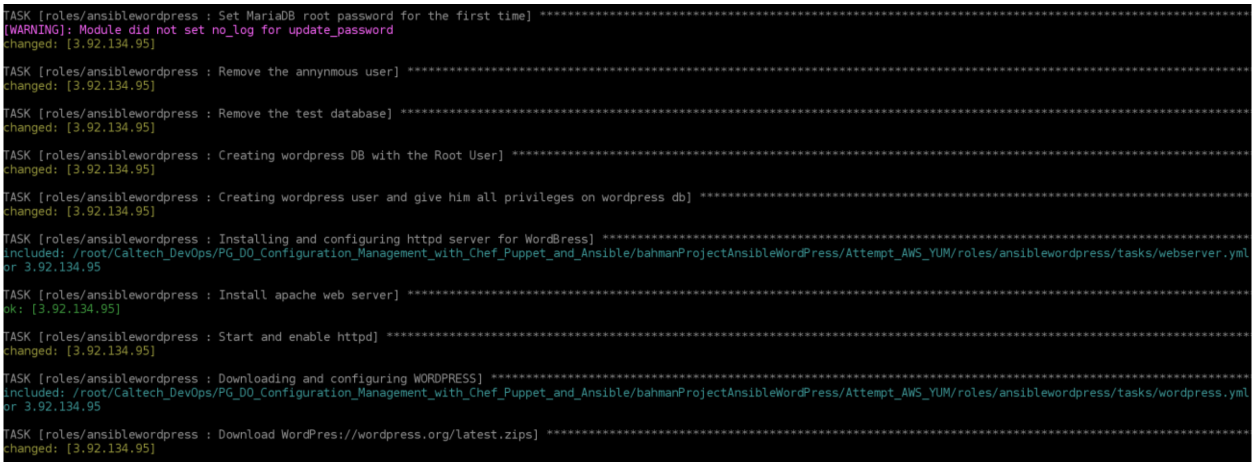
```
[root@ip-172-31-89-90 Attempt_AWS_YUM]# ansible-playbook wordpressplaybook.yml

PLAY [wordpress] ***************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************
[WARNING]: Platform linux on host 3.92.134.95 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change
this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [3.92.134.95]

TASK [roles/ansiblewordpress : bahman linux extra] *****************************************************************************
changed: [3.92.134.95]

TASK [roles/ansiblewordpress : Installing epel-release and yum-utils to download updated version of PHP] ***********************
[DEPRECATION WARNING]: Invoking "yum" only once while using a loop via squash_actions is deprecated. Instead of using a loop to supply multiple items and specifying `pkg: "{{ item
}}"`, please use `pkg: ['epel-release', 'yum-utils', 'http://rpms.remirepo.net/enterprise/remi-release-7.rpm']` and remove the loop. This feature will be removed in version 2.11.
Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
changed: [3.92.134.95] => (item=[u'epel-release', u'yum-utils', u'http://rpms.remirepo.net/enterprise/remi-release-7.rpm'])

TASK [roles/ansiblewordpress : enabling the PHP 7.3 Remi repository] ***********************************************************
changed: [3.92.134.95]

TASK [roles/ansiblewordpress : Installing PHP and other pkgs] ******************************************************************
[DEPRECATION WARNING]: Invoking "yum" only once while using a loop via squash_actions is deprecated. Instead of using a loop to supply multiple items and specifying `pkg: "{{ item
}}"`, please use `pkg: ['php', 'php-common', 'php-mysql', 'php-gd', 'php-xml', 'php-mbstring', 'php-xmlrpc', 'unzip', 'wget']` and remove the loop. This feature will be removed in
version 2.11. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
changed: [3.92.134.95] => (item=[u'php', u'php-common', u'php-mysql', u'php-gd', u'php-xml', u'php-mbstring', u'php-xmlrpc', u'unzip', u'wget'])

TASK [roles/ansiblewordpress : Installing and Configuring MariaDB and MariaDB-CLient For WordBress] ****************************
included: /root/Caltech_DevOps/PG_DO_Configuration_Management_with_Chef_Puppet_and_Ansible/bahmanProjectAnsibleWordPress/Attempt_AWS_YUM/roles/ansiblewordpress/tasks/database.yml f
r 3.92.134.95
```



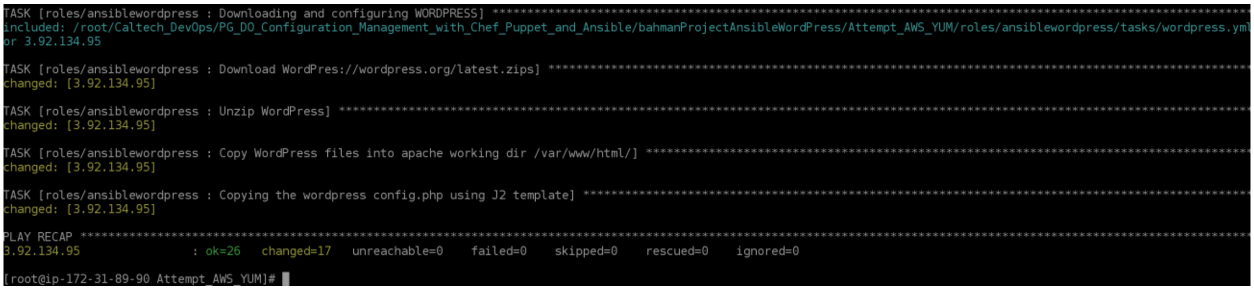
```
TASK [roles/ansiblewordpress : Installing and Configuring MariaDB and MariaDB-CLient For WordBress] ****************************
included: /root/Caltech_DevOps/PG_DO_Configuration_Management_with_Chef_Puppet_and_Ansible/bahmanProjectAnsibleWordPress/Attempt_AWS_YUM/roles/ansiblewordpress/tasks/database.yml
r 3.92.134.95

TASK [roles/ansiblewordpress : Install MariaDB-Server] *************************************************************************
changed: [3.92.134.95]

TASK [roles/ansiblewordpress : Install MariaDB-client] *************************************************************************
ok: [3.92.134.95]

TASK [roles/ansiblewordpress : Install MySQL-python pkg] ***********************************************************************
changed: [3.92.134.95]

TASK [roles/ansiblewordpress : Start and enable MariaDB-Server] ****************************************************************
changed: [3.92.134.95]

TASK [roles/ansiblewordpress : Pause to build database cache] ******************************************************************
Pausing for 11 seconds
(ctrl+C then 'C' = continue early, ctrl+C then 'A' = abort)
ok: [3.92.134.95]

TASK [roles/ansiblewordpress : Check if the root password is previously set or NOT] ********************************************
ok: [3.92.134.95]
```



```
TASK [roles/ansiblewordpress : Check if the root password is previously set or NOT] ********************************************
ok: [3.92.134.95]

TASK [roles/ansiblewordpress : debug] ******************************************************************************************
ok: [3.92.134.95] => {
    "root_pwd_check": {
        "changed": false,
        "cmd": "mysqladmin -u root status",
        "delta": "0:00:00.043620",
        "end": "2021-12-20 07:11:03.964121",
        "failed": false,
        "failed_when_result": false,
        "rc": 0,
        "start": "2021-12-20 07:11:03.920501",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "Uptime: 13  Threads: 1  Questions: 2  Slow queries: 0  Opens: 0  Flush tables: 2  Open tables: 26  Queries per second avg: 0.153",
        "stdout_lines": [
            "Uptime: 13  Threads: 1  Questions: 2  Slow queries: 0  Opens: 0  Flush tables: 2  Open tables: 26  Queries per second avg: 0.153"
        ]
    }
}

TASK [roles/ansiblewordpress : Set MariaDB root password for the first time] ***************************************************
[WARNING]: Module did not set no_log for update_password
changed: [3.92.134.95]

TASK [roles/ansiblewordpress : Remove the annynmous user] **********************************************************************
changed: [3.92.134.95]
```

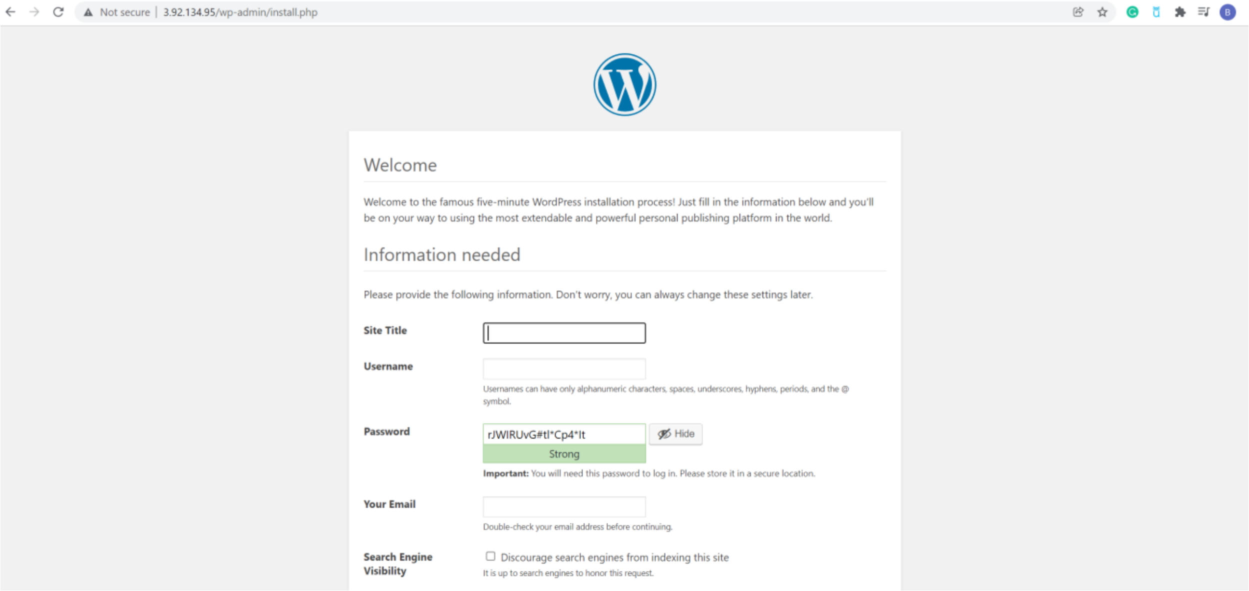


```
TASK [roles/ansiblewordpress : Set MariaDB root password for the first time] ****************************************************************
[WARNING]: Module did not set no_log for update_password
changed: [3.92.134.95]

TASK [roles/ansiblewordpress : Remove the annynmous user] ***********************************************************************************
changed: [3.92.134.95]

TASK [roles/ansiblewordpress : Remove the test database] ************************************************************************************
changed: [3.92.134.95]

TASK [roles/ansiblewordpress : Creating wordpress DB with the Root User] ********************************************************************
changed: [3.92.134.95]

TASK [roles/ansiblewordpress : Creating wordpress user and give him all privileges on wordpress db] *****************************************
changed: [3.92.134.95]

TASK [roles/ansiblewordpress : Installing and configuring httpd server for WordBress] *******************************************************
included: /root/Caltech_DevOps/PG_DO_Configuration_Management_with_Chef_Puppet_and_Ansible/bahmanProjectAnsibleWordPress/Attempt_AWS_YUM/roles/ansiblewordpress/tasks/webserver.yml
or 3.92.134.95

TASK [roles/ansiblewordpress : Install apache web server] ***********************************************************************************
ok: [3.92.134.95]

TASK [roles/ansiblewordpress : Start and enable httpd] **************************************************************************************
changed: [3.92.134.95]

TASK [roles/ansiblewordpress : Downloading and configuring WORDPRESS] ***********************************************************************
included: /root/Caltech_DevOps/PG_DO_Configuration_Management_with_Chef_Puppet_and_Ansible/bahmanProjectAnsibleWordPress/Attempt_AWS_YUM/roles/ansiblewordpress/tasks/wordpress.yml
or 3.92.134.95

TASK [roles/ansiblewordpress : Download WordPres://wordpress.org/latest.zips] ***************************************************************
changed: [3.92.134.95]
```



```
TASK [roles/ansiblewordpress : Downloading and configuring WORDPRESS] ***********************************************************************
included: /root/Caltech_DevOps/PG_DO_Configuration_Management_with_Chef_Puppet_and_Ansible/bahmanProjectAnsibleWordPress/Attempt_AWS_YUM/roles/ansiblewordpress/tasks/wordpress.yml
or 3.92.134.95

TASK [roles/ansiblewordpress : Download WordPres://wordpress.org/latest.zips] ***************************************************************
changed: [3.92.134.95]

TASK [roles/ansiblewordpress : Unzip WordPress] *********************************************************************************************
changed: [3.92.134.95]

TASK [roles/ansiblewordpress : Copy WordPress files into apache working dir /var/www/html/] *************************************************
changed: [3.92.134.95]

TASK [roles/ansiblewordpress : Copying the wordpress config.php using J2 template] **********************************************************
changed: [3.92.134.95]

PLAY RECAP **********************************************************************************************************************************
3.92.134.95                : ok=26   changed=17   unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[root@ip-172-31-89-90 Attempt_AWS_YUM]# █
```

14. Check WordPress on worker node:



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password rJWIRUvG#tl*Cp4*lt

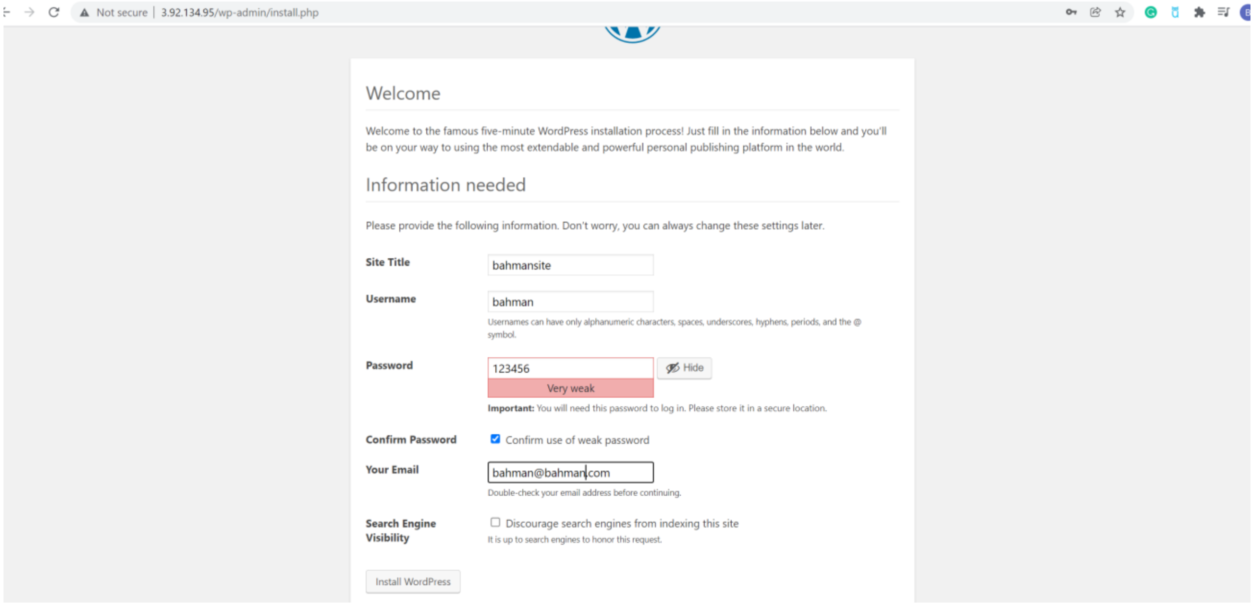
Important: You will need this password to log in. Please store it in a secure location.

Your Email

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.



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title bahmansite

Username bahman

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password 123456 Hide

Very weak

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

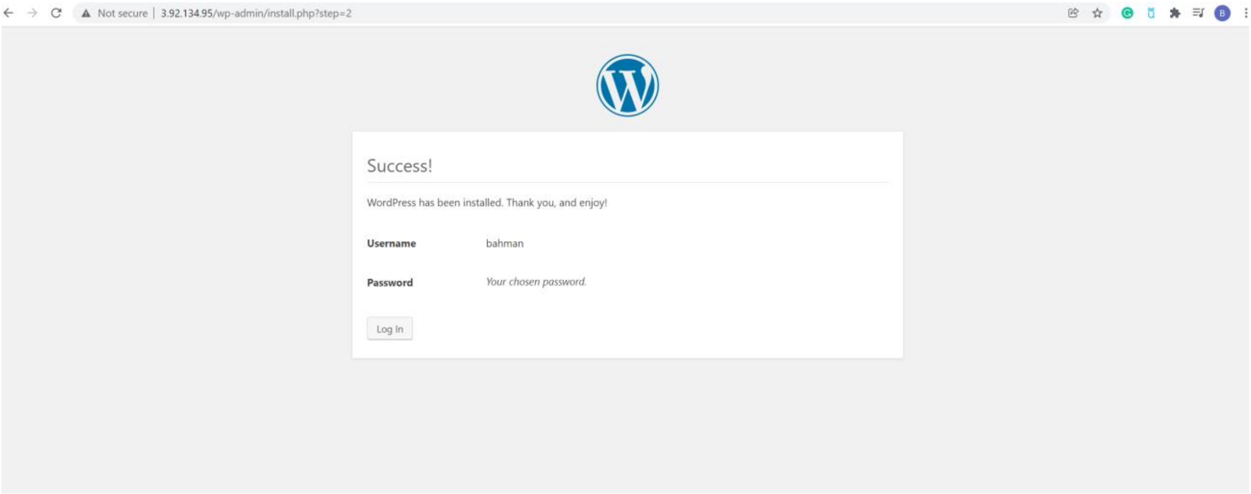
Confirm Password ☑ Confirm use of weak password

Your Email bahman@bahman.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.

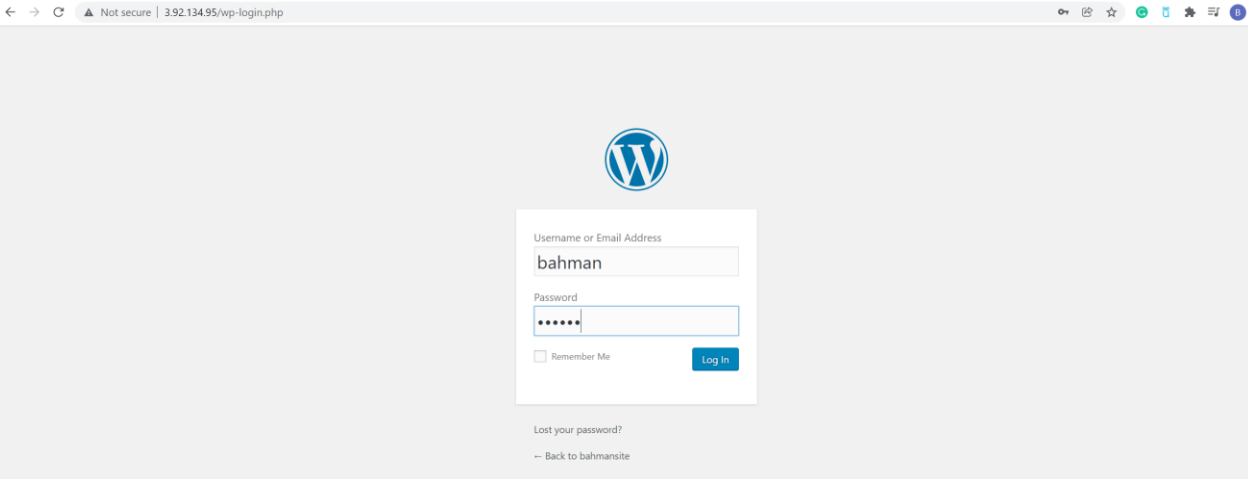


Success!

WordPress has been installed. Thank you, and enjoy!

Username bahman

Password *Your chosen password.*



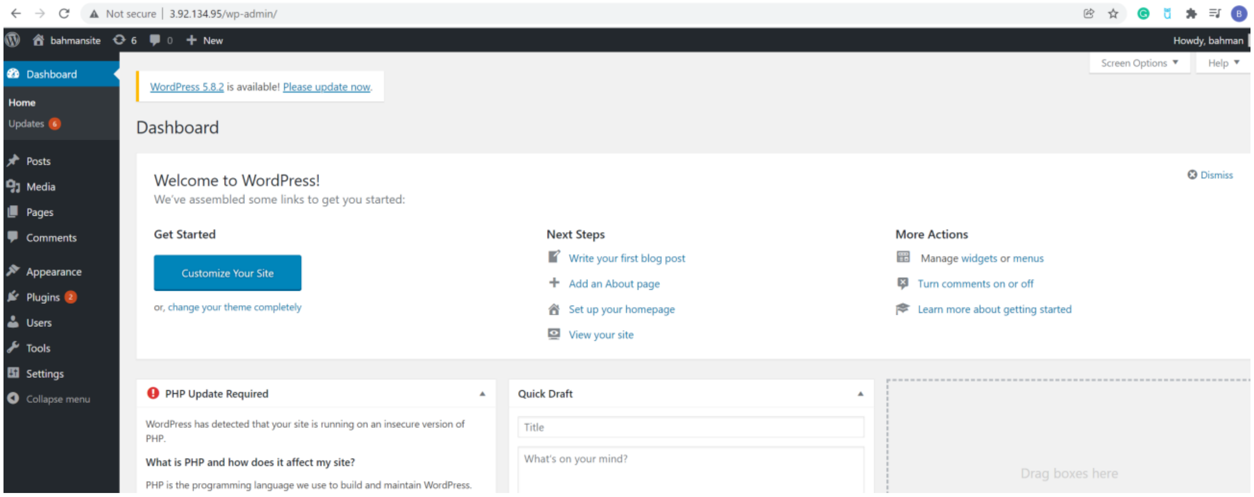
Username or Email Address

bahman

Password

••••••

□ Remember Me



WordPress 5.8.2 is available! Please update now.

Dashboard

Welcome to WordPress!

We've assembled some links to get you started:

Get Started

or, change your theme completely

Next Steps

- Write your first blog post
- Add an About page
- Set up your homepage
- View your site

More Actions

- Manage widgets or menus
- Turn comments on or off
- Learn more about getting started

PHP Update Required

WordPress has detected that your site is running on an insecure version of PHP.

What is PHP and how does it affect my site?

PHP is the programming language we use to build and maintain WordPress.

Quick Draft

Title

What's on your mind?