



## CentOS 7 – LAMP Stack

Ansible Playbook for CentOS 7 to stand up a LAMP server with phpMyAdmin

Filesystem Layout:

```
site/  
site/lamp-server.yml  
site/roles/  
site/roles/apache-server/  
site/roles/php-stack/  
site/roles/mariadb-server/  
site/roles/phpmyadmin/
```

This playbook consists of several roles. While they could all be combined into a single monolithic role I think it best to have them separated for reuse. If you're only wanting to stand up an Apache web server, use only that role, etc.

```
1 ansible-galaxy init apache-server  
2 ansible-galaxy init php-stack  
3 ansible-galaxy init mariadb-server  
4 ansible-galaxy init phpmyadmin
```

Role: apache-server

This role only consists of tasks to install and start apache on the server.

September 17, 2017

By ehoffman0811

In [ansible](#), [linux](#)

Search ...

### Recent Posts

Confluence Stack  
OpenSSL Self Signed Cert with SAN  
Check\_MK and PHP7  
Oracle Linux 7 – Local Repository  
CentOS 7 – LAMP Stack

### Archives

June 2018  
March 2018  
December 2017  
September 2017  
August 2017  
June 2017  
May 2017  
April 2017  
March 2017

### Categories

ansible  
automation

docker  
linux  
macos  
windows

**Meta**

Log in  
Entries feed  
Comments feed  
WordPress.org

Role: php-stack

Again, only tasks here with the basics for working php that will support most applications.

```
1 ---
2 - name: Install php packages
3   yum: name={{ item }} state=present
4   with_items:
5     - php
6     - php-gd
7     - php-mysql
8     - php-mbstring
9     - php-process
10    - php-xmlrpc
11    - php-xml
12    - libxslt
13    - php-mcrypt
```

Role: mariadb-server

This role is a bit more complex as MySQL, ahem MariaDB requires a bit of configuration to work properly and securely.

```
1 ---
2 # handlers file for mariadb-server
3 - name: restart firewalld
4   systemd:
5     name: firewalld
6     state: restarted
7
8 - name: restart httpd
9   systemd:
10    name: httpd
11    state: restarted
12
13 - name: restart mysql
14   systemd:
15    name: mariadb
16    state: restarted
```

```
1 ---
2 # tasks file for mariadb-server
3 - name: install mariadb-server
4   yum: name={{ item }} state=present
5   with_items:
6     - mariadb
7     - mariadb-server
8     - MySQL-python
9     - mysql-connector-python
10
11 - name: start mysqld
12   systemd:
13    name: mariadb
14    state: started
15    enabled: true
16
17 - name: open port 3306
18   firewalld:
19     port: 3306/tcp
20     permanent: true
21     state: enabled
22     immediate: yes
```

```

23  notify: restart firewalld
24
25  - name: check if root password for mariadb is set
26    shell: >
27      mysqladmin -u root status
28    changed_when: false
29    failed_when: false
30    register: root_pwd_check
31
32  - name: set mariadb root password for first time
33    mysql_user:
34      name: root
35      password: "{{ mysql_root_password }}"
36      host_all: yes
37      state: present
38    when: root_pwd_check.rc == 0
39
40  - name: flush privileges
41    command: 'mysql -ne "{{ item }}" -uroot -p "{{ mysql_root_password }}"'
42    with_items:
43      - FLUSH PRIVILEGES
44    changed_when: False
45
46  - name: Remove anonymous users
47    mysql_user:
48      name: ''
49      host_all: yes
50      login_user: root
51      login_password: "{{ mysql_root_password }}"
52      state: absent
53
54  - name: Remove test database and access to it
55    mysql_db:
56      name: test
57      login_user: root
58      login_password: "{{ mysql_root_password }}"
59      state: absent
60
61  - name: Disallow root login remotely
62    command: 'mysql -ne "{{ item }}" -uroot -p "{{ mysql_root_password }}"'
63    with_items:
64      - DELETE FROM mysql.user WHERE User='root' AND Host NOT IN ('localhost', '127.0.0.1', ':::
65    changed_when: False
66
67  - name: flush privileges
68    command: 'mysql -ne "{{ item }}" -uroot -p "{{ mysql_root_password }}"'
69    with_items:
70      - FLUSH PRIVILEGES
71    changed_when: False

```

Since the above tasks include variables for the mysql root user, we'll create an ansible-vault to keep them safe from prying eyes.

```

1  ansible-vault encrypt mariadb-server/vars/main.yml
2  New Vault password:
3  Confirm New Vault password:

```

Now we can edit the file and add the variable for our mysql root user

```

1  ansible-vault edit mariadb-server/vars/main.yml
2  Vault password:

```

```

1  ---
2  # vars file for mariadb-server
3  mysql_root_password: SuperSecretHardToBreakPassword

```

And you can prove to yourself it is secure by issuing a cat command to view the contents:

```

1  cat mariadb-server/vars/main.yml
2  $ANSIBLE_VAULT;1.1;AES256
3  30666631393162386366396263616436653465616461373130633830333165393865616432616136
4  353461636235653533373363338626530346539633537390a336437306365646330383632633538
5  62613xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx3138313465663233353739623535
6  3130356464306336360a313563336237656138313934303264633239393932343362383830653438
7  32633634316636393064666531383265643430343239616531393838666338633338623462313336
8  64346466323836623263363464646138363862653364653835363937313966346139363730373336
9  34396634373437303133313761373537323030313937313638663532336564656339376134613032
10 32636338316164373265

```

Role: phpmyadmin

This role consists of two tasks, a handler and a single config file to copy once installed to allow access from outside of the LAMP server.

```
1 ---
2 - name: restart httpd
3   systemd:
4     name: httpd
5     state: restarted
```

This is the phpMyAdmin.conf Apache config file. The only changes to the shipped file are adding private subnets to the Require ip and Allow ip spaces.

```
1 # phpMyAdmin - Web based MySQL browser written in php
2 #
3 # Allows only localhost by default
4 #
5 # But allowing phpMyAdmin to anyone other than localhost should be considered
6 # dangerous unless properly secured by SSL
7
8 Alias /phpMyAdmin /usr/share/phpMyAdmin
9 Alias /phpmyadmin /usr/share/phpMyAdmin
10
11 <Directory /usr/share/phpMyAdmin/>
12     AddDefaultCharset UTF-8
13
14     <IfModule mod_authz_core.c>
15         # Apache 2.4
16         <RequireAny>
17             Require ip 127.0.0.1
18             Require ip 192.168.1.0/24
19             Require ip 10.0.0.0/8
20             Require ip 172.16.0.0/16
21             Require ip ::1
22         </RequireAny>
23     </IfModule>
24     <IfModule !mod_authz_core.c>
25         # Apache 2.2
26         Order Deny,Allow
27         Deny from All
28         Allow from 127.0.0.1
29         Allow from 192.168.1.0/24
30         Allow from 10.0.0.0/8
31         Allow from 172.16.0.0/16
32         Allow from ::1
33     </IfModule>
34 </Directory>
35
36 <Directory /usr/share/phpMyAdmin/setup/>
37     <IfModule mod_authz_core.c>
38         # Apache 2.4
39         <RequireAny>
40             Require ip 127.0.0.1
41             Require ip 192.168.1.0/24
42             Require ip 10.0.0.0/8
43             Require ip 172.16.0.0/16
44             Require ip ::1
45         </RequireAny>
46     </IfModule>
47     <IfModule !mod_authz_core.c>
48         # Apache 2.2
49         Order Deny,Allow
50         Deny from All
51         Allow from 127.0.0.1
52         Allow from 192.168.1.0/24
53         Allow from 10.0.0.0/8
54         Allow from 172.16.0.0/16
55         Allow from ::1
```

```
56     </IfModule>
57 </Directory>
58
59 # These directories do not require access over HTTP - taken from the original
60 # phpMyAdmin upstream tarball
61 #
62 <Directory /usr/share/phpMyAdmin/libraries/>
63     Order Deny,Allow
64     Deny from All
65     Allow from None
66 </Directory>
67
68 <Directory /usr/share/phpMyAdmin/setup/lib/>
69     Order Deny,Allow
70     Deny from All
71     Allow from None
72 </Directory>
73
74 <Directory /usr/share/phpMyAdmin/setup/frames/>
75     Order Deny,Allow
76     Deny from All
77     Allow from None
78 </Directory>
79
80 # This configuration prevents mod_security at phpMyAdmin directories from
81 # filtering SQL etc. This may break your mod_security implementation.
82 #
83 <IfModule mod_security.c>
84     <Directory /usr/share/phpMyAdmin/>
85         SecRuleInheritance Off
86     </Directory>
87 </IfModule>
```

And now we can put it all together into a playbook to build out the LAMP server.

```
1 ---
2 - hosts: lamp-servers
3   roles:
4     - apache-server
5     - php-stack
6     - mariadb-server
7     - phpmyadmin
8   become: yes
```

Now, to deploy we issue the following while keeping in mind we have a vault to decrypt at run time:

```
./end
```

[Previous Post](#)

[Next Post](#)