**W**[**hat Does this Playbook Do?**](https://www.digitalocean.com/community/tutorials/how-to-use-ansible-to-install-and-set-up-lamp-on-ubuntu-18-04#what-does-this-playbook-do)

This Ansible playbook provides an alternative to manually running through the procedure outlined in our guide on [How To Install Linux, Apache, MySQL and PHP (LAMP) on Ubuntu 18.04](https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-ubuntu-18-04).

Running this playbook will perform the following actions on your Ansible hosts:

1. Install aptitude, which is preferred by Ansible as an alternative to the apt package manager.
2. Install the required LAMP packages.
3. Create a new Apache VirtualHost and set up a dedicated document root for that.
4. Enable the new VirtualHost.
5. Disable the default Apache website, when the **disable\_default** variable is set to true.
6. Set the password for the MySQL **root** user.
7. Remove anonymous MySQL accounts and the test database.
8. Set up UFW to allow HTTP traffic on the configured port (80 by default).
9. Set up a PHP test script using the provided template.

Once the playbook has finished running, you will have a web PHP environment running on top of Apache, based on the options you defined within your configuration variables.

[**How to Use this Playbook**](https://www.digitalocean.com/community/tutorials/how-to-use-ansible-to-install-and-set-up-lamp-on-ubuntu-18-04#how-to-use-this-playbook)

The first thing we need to do is obtain the LAMP playbook and its dependencies from the [do-community/ansible-playbooks](https://github.com/do-community/ansible-playbooks) repository. We need to clone this repository to a local folder inside the Ansible Control Node.

In case you have cloned this repository before while following a different guide, access your existing ansible-playbooks copy and run a git pull command to make sure you have updated contents:

1. cd ~/ansible-playbooks
2. git pull

Copy

If this is your first time using the do-community/ansible-playbooks repository, you should start by cloning the repository to your home folder with:

1. cd ~
2. git clone https://github.com/do-community/ansible-playbooks.git
3. cd ansible-playbooks

Copy

The files we’re interested in are located inside the lamp\_ubuntu1804 folder, which has the following structure:

lamp\_ubuntu1804

├── files

│ ├── apache.conf.j2

│ └── info.php.j2

├── vars

│ └── default.yml

├── playbook.yml

└── readme.md

Here is what each of these files are:

* files/info.php.j2: Template file for setting up a PHP test page on the web server’s root
* files/apache.conf.j2: Template file for setting up the Apache VirtualHost.
* vars/default.yml: Variable file for customizing playbook settings.
* playbook.yml: The playbook file, containing the tasks to be executed on the remote server(s).
* readme.md: A text file containing information about this playbook.

We’ll edit the playbook’s variable file to customize the configurations of both MySQL and Apache. Access the lamp\_ubuntu1804 directory and open the vars/default.yml file using your command line editor of choice:

1. cd lamp\_ubuntu1804
2. nano vars/default.yml

Copy

This file contains a few variables that require your attention:

vars/default.yml

---

mysql\_root\_password: "mysql\_root\_password"

app\_user: "sammy"

http\_host: "your\_domain"

http\_conf: "your\_domain.conf"

http\_port: "80"

disable\_default: **true**

Copy

The following list contains a brief explanation of each of these variables and how you might want to change them:

* mysql\_root\_password: The desired password for the **root** MySQL account.
* app\_user: A remote non-root user on the Ansible host that will be set as the owner of the application files.
* http\_host: Your domain name.
* http\_conf: The name of the configuration file that will be created within Apache.
* http\_port: HTTP port for this virtual host, where 80 is the default.
* disable\_default: Whether or not to disable the default website that comes with Apache.

Once you’re done updating the variables inside vars/default.yml, save and close this file. If you used nano, do so by pressing CTRL + X, Y, then ENTER.

You’re now ready to run this playbook on one or more servers. Most playbooks are configured to be executed on every server in your inventory, by default. We can use the -l flag to make sure that only a subset of servers, or a single server, is affected by the playbook. We can also use the -u flag to specify which user on the remote server we’re using to connect and execute the playbook commands on the remote hosts.

To execute the playbook only on server1, connecting as sammy, you can use the following command:

1. ansible-playbook playbook.yml -l server1 -u sammy

Copy

You will get output similar to this:

Output

PLAY [all] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ok: [server1]

TASK [Install prerequisites] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ok: [server1] => (item=aptitude)

...

TASK [UFW - Allow HTTP on port 80] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [server1]

TASK [Sets Up PHP Info Page] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [server1]

RUNNING HANDLER [Reload Apache] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [server1]

RUNNING HANDLER [Restart Apache] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [server1]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

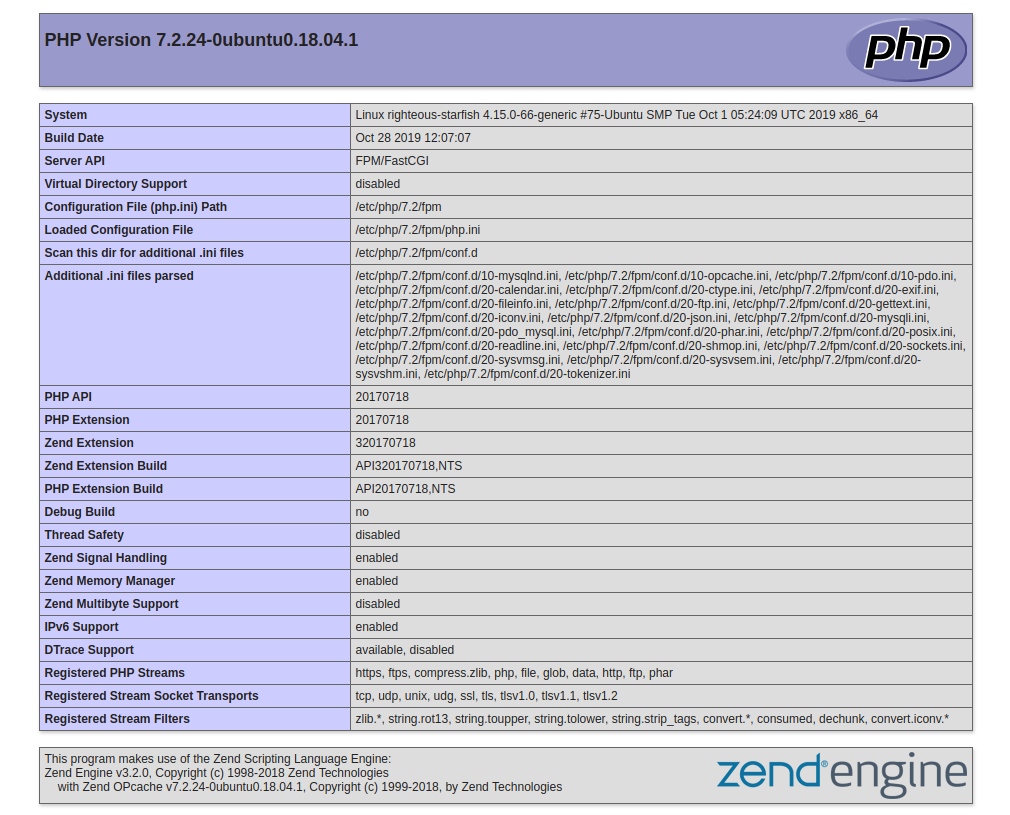
server1 : ok=15 changed=11 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

**Note**: For more information on how to run Ansible playbooks, check our [Ansible Cheat Sheet Guide](https://www.digitalocean.com/community/tutorials/how-to-use-ansible-cheat-sheet-guide).

When the playbook is finished running, go to your web browser and access the host or IP address of the server, as configured in the playbook variables, followed by /info.php:

http://server\_host\_or\_IP/info.php

You will see a page like this:



Because this page contains sensitive information about your PHP environment, it is recommended that you remove it from the server by running an rm -f /var/www/info.php command once you have finished setting it up.

[**The Playbook Contents**](https://www.digitalocean.com/community/tutorials/how-to-use-ansible-to-install-and-set-up-lamp-on-ubuntu-18-04#the-playbook-contents)

You can find the LAMP server setup featured in this tutorial in the [lamp\_ubuntu1804](https://github.com/do-community/ansible-playbooks/tree/master/apache_ubuntu1804) folder inside the [DigitalOcean Community Playbooks](https://github.com/do-community/ansible-playbooks) repository. To copy or download the script contents directly, click the **Raw** button towards the top of each script.

The full contents of the playbook as well as its associated files are also included here for your convenience.

[**vars/default.yml**](https://www.digitalocean.com/community/tutorials/how-to-use-ansible-to-install-and-set-up-lamp-on-ubuntu-18-04#vars-default-yml)

The default.yml variable file contains values that will be used within the playbook tasks, such as the password for the MySQL **root** account and the domain name to configure within Apache.

vars/default.yml

---

mysql\_root\_password: "mysql\_root\_password"

app\_user: "sammy"

http\_host: "your\_domain"

http\_conf: "your\_domain.conf"

http\_port: "80"

disable\_default: **true**

Copy

[**files/apache.conf.j2**](https://www.digitalocean.com/community/tutorials/how-to-use-ansible-to-install-and-set-up-lamp-on-ubuntu-18-04#files-apache-conf-j2)

The apache.conf.j2 file is a [Jinja 2](https://jinja.palletsprojects.com/en/2.10.x/) template file that configures a new Apache VirtualHost. The variables used within this template are defined in the vars/default.yml variable file.

files/apache.conf.j2

<VirtualHost \*:{{ http\_port }}>

ServerAdmin webmaster@localhost

ServerName {{ http\_host }}

ServerAlias www.{{ http\_host }}

DocumentRoot /var/www/{{ http\_host }}

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

<Directory /var/www/{{ http\_host }}>

Options -Indexes

</Directory>

<IfModule mod\_dir.c>

DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm

</IfModule>

</VirtualHost>

[**files/info.php.j2**](https://www.digitalocean.com/community/tutorials/how-to-use-ansible-to-install-and-set-up-lamp-on-ubuntu-18-04#files-info-php-j2)

The info.php.j2 file is another Jinja template, used to set up a test PHP script in the document root of the newly configured LAMP server.

files/info.php.j2

<?php

phpinfo();

[**playbook.yml**](https://www.digitalocean.com/community/tutorials/how-to-use-ansible-to-install-and-set-up-lamp-on-ubuntu-18-04#playbook-yml)

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 (all), 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={{ item }} update\_cache=yes state=latest force\_apt\_get=yes

loop: [ 'aptitude' ]

#Apache Configuration

- name: Install LAMP Packages

apt: name={{ item }} update\_cache=yes state=latest

loop: [ 'apache2', 'mysql-server', 'python3-pymysql', 'php', 'php-mysql', 'libapache2-mod-php' ]

- name: Create document root

file:

path: "/var/www/{{ http\_host }}"

state: directory

owner: "{{ app\_user }}"

mode: '0755'

- name: Set up Apache virtualhost

template:

src: "files/apache.conf.j2"

dest: "/etc/apache2/sites-available/{{ http\_conf }}"

notify: Reload Apache

- name: Enable new site

shell: /usr/sbin/a2ensite {{ http\_conf }}

notify: Reload Apache

- name: Disable default Apache site

shell: /usr/sbin/a2dissite 000-default.conf

when: disable\_default

notify: Reload Apache

# MySQL Configuration

- name: Sets the root password

mysql\_user:

name: root

password: "{{ mysql\_root\_password }}"

login\_unix\_socket: /var/run/mysqld/mysqld.sock

- name: Removes all anonymous user accounts

mysql\_user:

name: ''

host\_all: yes

state: absent

login\_user: root

login\_password: "{{ mysql\_root\_password }}"

- name: Removes the MySQL test database

mysql\_db:

name: test

state: absent

login\_user: root

login\_password: "{{ mysql\_root\_password }}"

# UFW Configuration

- name: "UFW - Allow HTTP on port {{ http\_port }}"

ufw:

rule: allow

port: "{{ http\_port }}"

proto: tcp

# PHP Info Page

- name: Sets Up PHP Info Page

template:

src: "files/info.php.j2"

dest: "/var/www/{{ http\_host }}/info.php"

handlers:

- name: Reload Apache

service:

name: apache2

state: reloaded

- name: Restart Apache

service:

name: apache2

state: restarted

Copy

Feel free to modify these files to best suit your individual needs within your own workflow.

[**Conclusion**](https://www.digitalocean.com/community/tutorials/how-to-use-ansible-to-install-and-set-up-lamp-on-ubuntu-18-04#conclusion)

In this guide, we used Ansible to automate the process of installing and setting up a LAMP environment on a remote server. Because each individual typically has different needs when working with MySQL databases and users, we encourage you to check out the [official Ansible documentation](https://docs.ansible.com/ansible/latest/modules/mysql_user_module.html#mysql-user-module) for more information and use cases of the mysql\_user Ansible module.