

CLOUD AUTOMATION

Provisioning of Two Digital Ocean Droplets as WordPress Website
Platforms Using Ansible Automation Tool

CORY SEARCY

COURSE: PROJECT & PORTFOLIO 6



TABLE OF CONTENTS

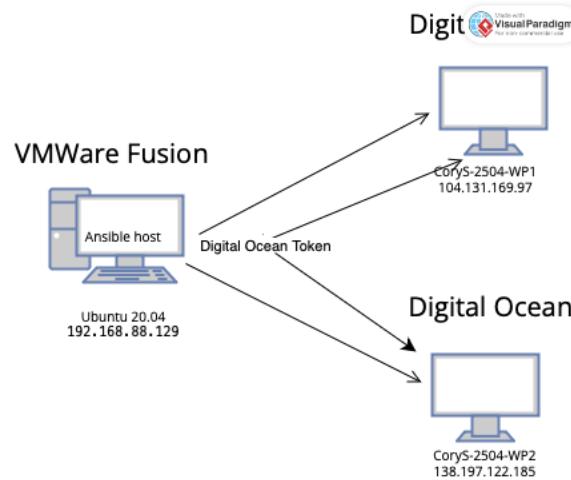
PROJECT SCOPE OVERVIEW	2
NETWORK TOPOLOGY DIAGRAM WITH IP ADDRESSES	2
CLOUD AUTOMATION PROCEDURAL INSTRUCTIONS	2-14
FINAL SCRIPT	14-18
LESSONS LEARNED	18
APPENDIX A DOCUMENT ERRORS & SOLUTIONS	18

Scope of Work

Project Scope Overview

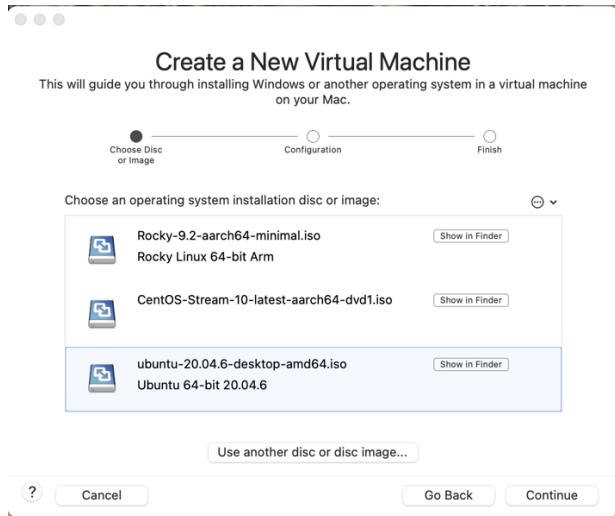
Provision and configure two Digital Ocean droplets as independent WordPress websites using Ansible for infrastructure automation. This project aims to ensure fast, repeatable, and reliable deployments with minimal manual intervention.

NETWORK TOPOLOGY DIAGRAM WITH IP ADDRESSES



CLOUD AUTOMATION PROCEDURAL INSTRUCTIONS

Create a new Ubuntu VM using VMware.



- Install and configure Ubuntu Server.
- Now log in to the terminal on the Ubuntu VM.

```
cory@cory-virtual-machine:~$
```

- Run sudo apt update & sudo apt upgrade -y

```
cory@cory-virtual-machine:~$ sudo apt update
Reading package lists... Done
E: Could not get lock /var/lib/apt/lists/lock. It is held by process 42388 (apt-get)
N: Be aware that removing the lock file is not a solution and may break your system.
E: Unable to lock directory /var/lib/apt/lists/
cory@cory-virtual-machine:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  ieee-data python3-argcomplete python3-dnspython python3-libcloud python3-netaddr
  python3-pycryptodome python3-requests-toolbelt python3-selinux python3-simplejson
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  linux-headers-6.8.0-58-generic linux-hwe-6.8-headers-6.8.0-58 linux-hwe-6.8-tools-6.8.0-58
```

- Now install ansible using

- sudo apt install ansible -y

```
done
cory@cory-virtual-machine:~$ sudo apt install ansible -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ansible is already the newest version (10.7.0-1ppa-jammy).
The following packages were automatically installed and are no longer required:
  ieee-data linux-headers-6.8.0-57-generic linux-hwe-6.8-headers-6.8.0-57
  linux-hwe-6.8-tools-6.8.0-57 linux-image-6.8.0-57-generic linux-modules-6.8.0-57-generic
  linux-modules-extra-6.8.0-57-generic linux-tools-6.8.0-57-generic python3-argcomplete
  python3-dnspython python3-libcloud python3-netaddr python3-pycryptodome python3-requests-tool
  python3-selinux python3-simplejson
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
cory@cory-virtual-machine:~$
```

- Verify installation of ansible

Run ansible --version

```
cory@cory-virtual-machine:~$ ansible --version
ansible [core 2.17.11]
  config file = /home/cory/ansible.cfg
  configured module search path = ['~home/cory/.ansible/plugins/modules', '/usr/share/ansible/p
  modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/cory/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Feb  4 2025, 14:57:36) [GCC 11.4.0] (/usr/bin/python3)
  jinja version = 3.0.3
  libyaml = True
cory@cory-virtual-machine:~$
```

- Install python

Run sudo apt install python3

```
cory@cory-virtual-machine:~$ sudo apt install python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.6-1~22.04.1).
python3 set to manually installed.
The following packages were automatically installed and are no longer required:
  ieee-data linux-headers-6.8.0-57-generic linux-hwe-6.8-headers-6.8.0-57
  linux-hwe-6.8-tools-6.8.0-57 linux-image-6.8.0-57-generic linux-modules-6.8.0-57-generic
  linux-modules-extra-6.8.0-57-generic linux-tools-6.8.0-57-generic python3-argcomplete
  python3-dnspython python3-libcloud python3-netaddr python3-pycryptodome python3-requests-tool
  python3-selinux python3-simplejson
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

- Next, we need to install python-pip

Run sudo apt install python-pip

- Now we must open the ansible.cfg file

```
cory@cory-virtual-machine:~$ sudo nano ansible.cfg
```

- Inside the ansible.cfg file add
hostfile=hosts

```
cory@cory-virtual-machine:~$ sudo apt install python-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  ieee-data javascript-common libexpat1-dev libjs-jquery libjs-sphinxdoc libjs-underscore
  libpython3-dev libpython3.10-dev linux-headers-6.8.0-57-generic linux-hwe-6.8.0-57-generic
  linux-image-6.8.0-57-generic linux-modules-6.8.0-57-generic
  LibreOffice Writer extra-6.8.0-57-generic linux-tools-6.8.0-57-generic python3-argcomplete python3-dnspython
  python3-libcloud python3-netaddr python3-pycryptodome python3-requests-tools
  python3-selinux python3-simplejson python3-wheel python3.10-dev zlib1g-dev
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libpython2.7-stdlib libpython2.7-minimal libpython2.7-stdlib python-pkg-resources python-setuptools

```

Save

```
GNU nano 6.2                               ansible.cfg
[defaults]
hostfile=hosts

[ Read 2 lines ]
^G Help          ^O Write Out   ^W Where Is   ^K Cut          ^T Execute   ^C Location   M-U Undo
^X Exit          ^R Read File   ^H Replace    ^U Paste        ^J Justify    ^V Go To Line M-E Redo
```

- Next, nano into the host file and add the following

[localhost]

localhost ansible_connection=local

```
GNU nano 6.2                               hosts
[localhost]
localhost ansible_connection=local
```

- Now we need to generate ssh key
Run ssh-keygen -t rsa
- Next run ssh-copy-id <root@192.168.88.129>
- Now cd to the ansible directory

Run cd /etc/ansible

```
cory@cory-virtual-machine:~$ cd /etc/ansible
cory@cory-virtual-machine:/etc/ansible$
```

- Here we need to create an ansible playbook to configure 2 digital ocean droplets
Run sudo nano droplet1.yml

```
cory@cory-virtual-machine:/etc/ansible$
```

- Now edit the Ansible playbook to create droplets.



```
GNU nano 6.2          droplet1.yml

- name: Create Digital Ocean Droplets
  hosts: localhost
  connection: local
  gather_facts: false

  vars:
    droplet_name_prefix: "CoryS-2504-WP"
    region: "nyc3"
    size: "s-1vcpu-1gb"
    image: "ubuntu-22-04-x64"
    ssh_key_name: "Cory-Searcy-2504"
    number_of_droplets: 4
    droplets:
      - CoryS-2504-WP1
      - CoryS-2504-WP2

  tasks:
    - name: Create SSH Key in Digital Ocean
      digital_ocean_sshkey:
        name: "{{ ssh_key_name }}"
        ssh_pub_key: "{{ lookup('file', '~/.ssh/id_rsa.pub') }}"
      register: ssh_key_result

    - name: Create Droplets
      digital_ocean_droplet:
        name: "{{ droplet_name_prefix }}{{ item }}"
        region: "{{ region }}"
        size: "{{ size }}"
        image: "{{ image }}"
        ssh_keys: "{{ ssh_key_result.data.ssh_key.id }}"
        unique_name: yes
        state: present
      loop: "{{ range(1, number_of_droplets + 1) | list }}"
      register: droplet_result

    - name: Wait for droplets to be created
      pause:
        seconds: 30

    - name: Add new droplets to host group
      add_host:
        name: "{{ item.networks.v4[0].ip_address }}"
        groups: droplets
      loop: "{{ droplets_info.data }}"
      when: item.networks is defined and item.networks.v4 | length > 0
```

- Run ansible-playbook -i inventory.ini droplet1.yml
- This will create the droplets.
- Create a directory to hold WordPress related files.
- Use mkdir -p lamp_wordpress/{group_vars,templates}

- Now use `cd lamp_wordpress` to switch to the new directory.

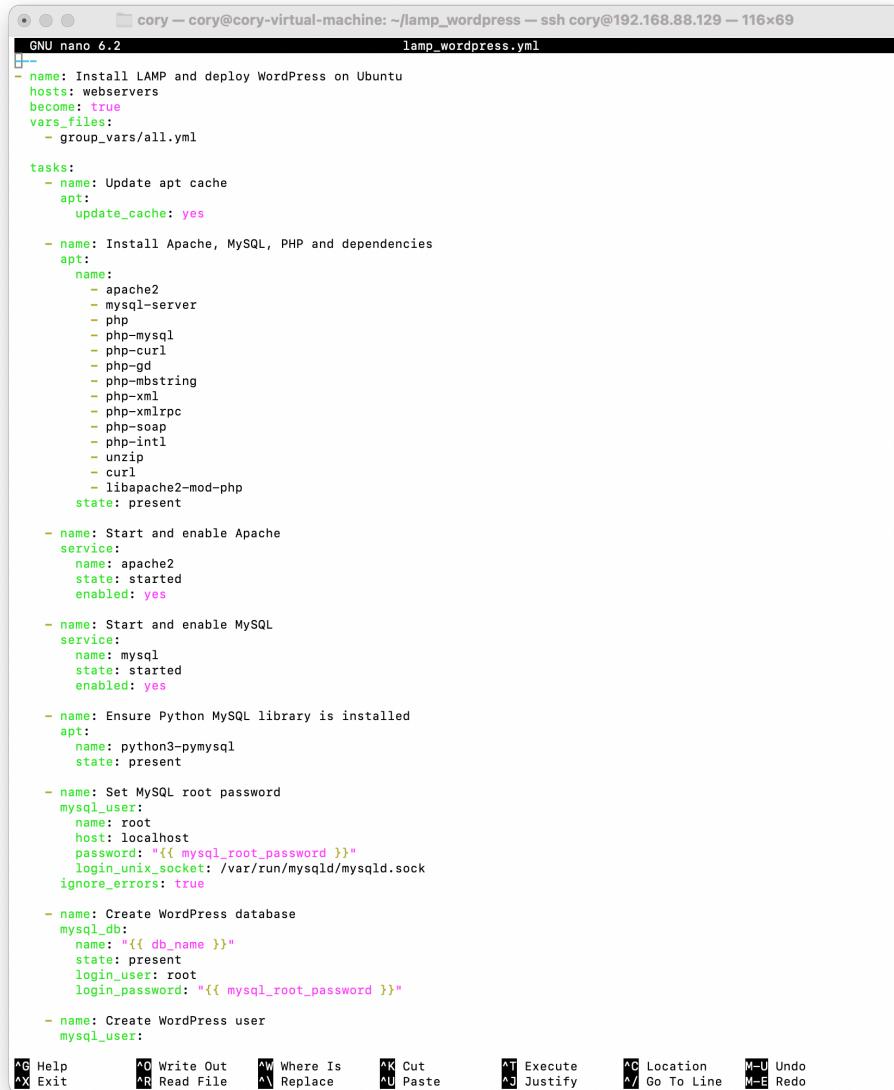
```
cory@cory-virtual-machine:~$ mkdir -p lamp_wordpress/{group_vars,templates}
cd lamp_wordpress
```

- Create `group_vars/all.yml` and add

```
GNU nano 6.2                                     group_vars/all.yml
mysql_root_password: 'Fullsail1!'
db_name: wordpress
db_user: wpuser
db_password: 'Fullsail1!'
wordpress_url: https://wordpress.org/latest.tar.gz
```

- Now, create the playbook by adding a lamp and WordPress to droplets.
- Run `sudo nano lamp_wordpress.yml`

- Edit the `lamp_wordpress.yml` playbook.



```
GNU nano 6.2                                     lamp_wordpress.yml
---
- name: Install LAMP and deploy WordPress on Ubuntu
  hosts: webservers
  become: true
  vars_files:
    - group_vars/all.yml

  tasks:
    - name: Update apt cache
      apt:
        update_cache: yes

    - name: Install Apache, MySQL, PHP and dependencies
      apt:
        name:
          - apache2
          - mysql-server
          - php
          - php-mysql
          - php-curl
          - php-gd
          - php-mbstring
          - php-xml
          - php-xmlrpc
          - php-soap
          - php-intl
          - unzip
          - curl
          - libapache2-mod-php
        state: present

    - name: Start and enable Apache
      service:
        name: apache2
        state: started
        enabled: yes

    - name: Start and enable MySQL
      service:
        name: mysql
        state: started
        enabled: yes

    - name: Ensure Python MySQL library is installed
      apt:
        name: python3-pymysql
        state: present

    - name: Set MySQL root password
      mysql_user:
        name: root
        host: localhost
        password: "{{ mysql_root_password }}"
        login_unix_socket: /var/run/mysqld/mysqld.sock
        ignore_errors: true

    - name: Create WordPress database
      mysql_db:
        name: "{{ db_name }}"
        state: present
        login_user: root
        login_password: "{{ mysql_root_password }}"

    - name: Create WordPress user
      mysql_user:
```

At the bottom of the terminal window, there is a menu bar with the following options: Help, Write Out, Where Is, Cut, Execute, Location, Undo, Exit, Read File, Replace, Paste, Justify, Go To Line, and Redo.

```

cory — cory@cory-virtual-machine: ~/lamp_wordpress — ssh cory@192.168.88.129 — 116x69
GNU nano 6.2                                     lamp_wordpress.yml
state: present

- name: Set MySQL root password
  mysql_user:
    name: root
    host: localhost
    password: "{{ mysql_root_password }}"
    login_unix_socket: /var/run/mysqld/mysqld.sock
  ignore_errors: true

- name: Create WordPress database
  mysql_db:
    name: "{{ db_name }}"
    state: present
    login_user: root
    login_password: "{{ mysql_root_password }}"

- name: Create WordPress user
  mysql_user:
    name: "{{ db_user }}"
    password: "{{ db_password }}"
    priv: "{{ db_name }}.*:ALL"
    host: '%'
    state: present
    login_user: root
    login_password: "{{ mysql_root_password }}"

- name: Download WordPress
  get_url:
    url: "{{ wordpress_url }}"
    dest: /tmp/latest.tar.gz

- name: Extract WordPress
  unarchive:
    src: /tmp/latest.tar.gz
    dest: /var/www/html/
    remote_src: yes

- name: Set permissions for WordPress directory
  file:
    path: /var/www/html/wordpress
    state: directory
    recurse: yes
    owner: www-data
    group: www-data
    mode: '0755'

- name: Copy wp-config.php
  template:
    src: wp-config.php.j2
    dest: /var/www/html/wordpress/wp-config.php
    owner: www-data
    group: www-data
    mode: '0644'

- name: Enable Apache rewrite module
  apache2_module:
    name: rewrite
    state: present

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

```

File menu: Help, Exit, Write Out, Read File, Where Is, Replace, Cut, Paste, Execute, Justify, Location, Go To Line, Undo, Redo.

-
-
- Create a WordPress config template
- Sudo nano templates/wp-config.php.j2
- `[cory@cory-virtual-machine:~/lamp_wordpress$ sudo nano templates/wp-config.php.j2`
- Use the following

```

GNU nano 6.2                                         templates/wp-config.php.j2
<?php
define( 'DB_NAME', '{{ db_name }}' );
define( 'DB_USER', '{{ db_user }}' );
define( 'DB_PASSWORD', '{{ db_password }}' );
define( 'DB_HOST', 'localhost' );
define( 'DB_CHARSET', 'utf8' );
define( 'DB_COLLATE', '' );

define('AUTH_KEY',         '+9+|{@DHvm`hKuW}chd/(ul  HcX` cn1_U<=RW@%D09]Hfvc-5h>_>FtpzAu<@[' );
define('SECURE_AUTH_KEY',  'ijf)mIuSV]Wqs]mOK`7~~iigB|{FH1+kqAy/VJsE*a|oi(la.~N(!!W|VG&Zf{7Q');
define('LOGGED_IN_KEY',    'c97BBPL6_%65a5(jRN-HD=D-b:Sw3!Sjvv=wfaJD~shn]8pX!q(m.ABY4E@h,Z{~');
define('NONCE_KEY',        '}t4#[?^0exsU~:ug1;@Zk%<Lmupd-va`A:,G:-0F(V*WgrVS/Rt;O~T-p+|tz-f');
define('AUTH_SALT',        'itV!*tkB5h0{k(HV[#~&6f==.IDxppR8Zssn!MBUEF-H,)TfK@Yldublhp01%p f');
define('SECURE_AUTH_SALT', 'Y/4,CfSh_7)l2w/~5}-]0t9N$/t2.qgcU0ipWd*@q1Ra&YG4$ p+PhbvPK4?MFy');
define('LOGGED_IN_SALT',   '0o_Z-3 `a?.78+FLI>MH=&gC-r&Kp*(Ag=.W$XH1[n0X;EbSn-zq gXsmT)@s9i4');
define('NONCE_SALT',       'FK}7W|B>)v$}b/_t6W<?b;<uJt?BO!/IGH$Z}n2-;[s0]8G1Srw<pbw2JgBrh,f-');

$table_prefix = 'wp_';
define( 'WP_DEBUG', false );

if ( ! defined( 'ABSPATH' ) ) {
    define( 'ABSPATH', dirname( __FILE__ ) . '/' );
}
require_once ABSPATH . 'wp-settings.php';

```

- Now we can run the playbook
- Use ansible-playbook -i inventory.ini lamp_wordpress.yml

```

cory — cory@cory-virtual-machine: ~/lamp_wordpress — ssh cory@192.168.88.129 — 116x69
ok: [CoryS-2504-WP2]

TASK [Start and enable MySQL] ****
ok: [CoryS-2504-WP1]
ok: [CoryS-2504-WP2]

TASK [Ensure Python MySQL library is installed] ****
ok: [CoryS-2504-WP2]
ok: [CoryS-2504-WP1]

TASK [Set MySQL root password] ****
fatal: [CoryS-2504-WP2]: FAILED! => {"changed": false, "msg": "unable to connect to database, check login_user and login_password are correct or /root/.my.cnf has the credentials. Exception message: (1045, \"Access denied for user 'root'@'localhost' (using password: NO)\\\")"}
...ignoring
fatal: [CoryS-2504-WP1]: FAILED! => {"changed": false, "msg": "unable to connect to database, check login_user and login_password are correct or /root/.my.cnf has the credentials. Exception message: (1045, \"Access denied for user 'root'@'localhost' (using password: NO)\\\")"}
...ignoring

TASK [Create WordPress database] ****
ok: [CoryS-2504-WP2]
ok: [CoryS-2504-WP1]

TASK [Create WordPress user] ****
[WARNING]: Option column_case_sensitive is not provided. The default is now false, so the column's name will be uppercased. The default will be changed to true in community.mysql 4.0.0.
ok: [CoryS-2504-WP2]
ok: [CoryS-2504-WP1]

TASK [Download WordPress] ****
ok: [CoryS-2504-WP1]
ok: [CoryS-2504-WP2]

TASK [Extract WordPress] ****
changed: [CoryS-2504-WP1]
changed: [CoryS-2504-WP2]

TASK [Set permissions for WordPress directory] ****
changed: [CoryS-2504-WP2]
changed: [CoryS-2504-WP1]

TASK [Copy wp-config.php] ****
changed: [CoryS-2504-WP1]
changed: [CoryS-2504-WP2]

TASK [Enable Apache rewrite module] ****
ok: [CoryS-2504-WP1]
ok: [CoryS-2504-WP2]

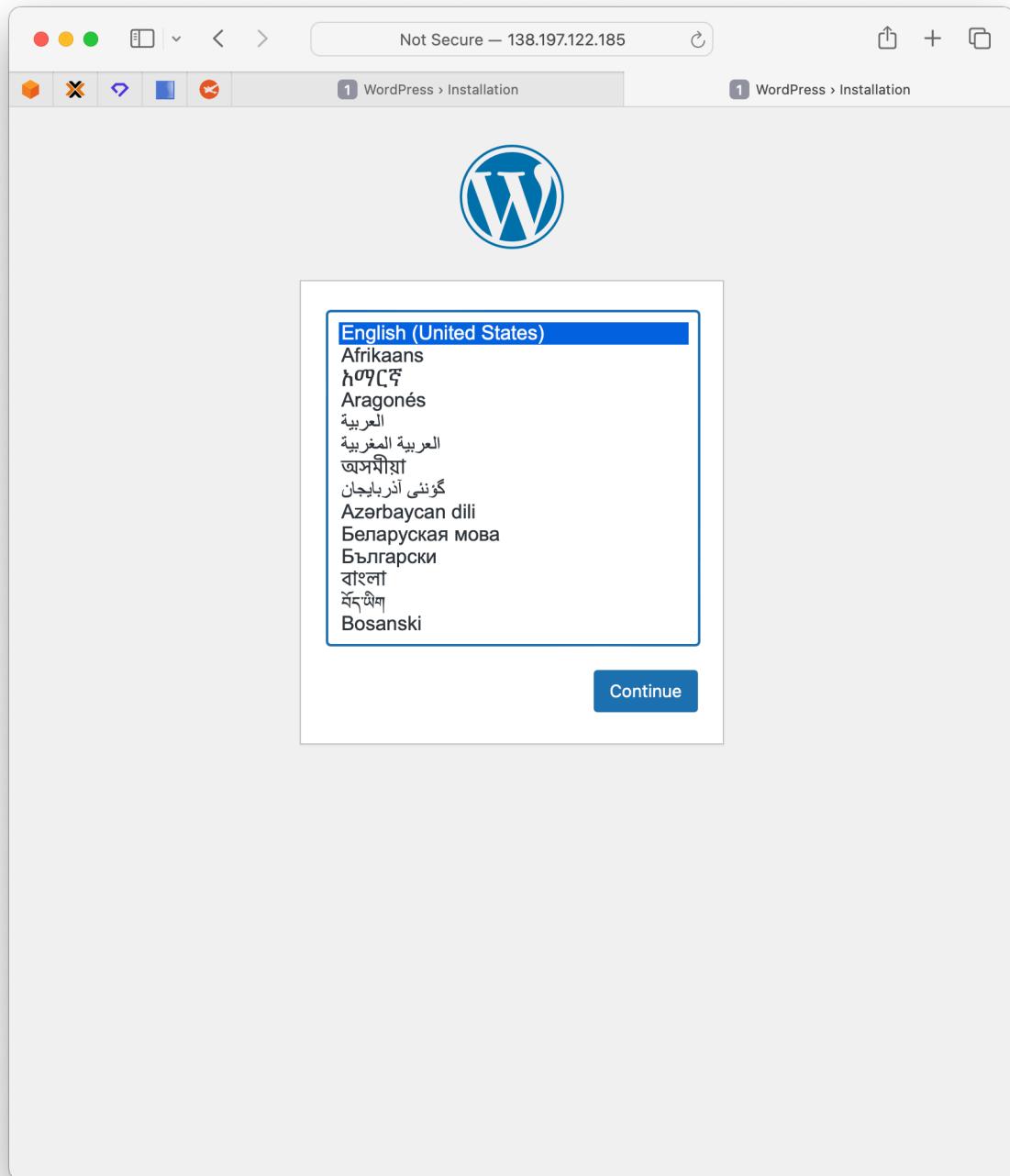
TASK [Restart Apache] ****
changed: [CoryS-2504-WP1]
changed: [CoryS-2504-WP2]

PLAY RECAP ****
CoryS-2504-WP1 : ok=15  changed=5    unreachable=0   failed=0    skipped=0   rescued=0   ignored=1
CoryS-2504-WP2 : ok=15  changed=5    unreachable=0   failed=0    skipped=0   rescued=0   ignored=1

```

- Now we must test our WordPress sites

- Use <http://138.197.122.185/wordpress/> to pull up WordPress for CoryS-2504-WP2



The screenshot shows the WordPress dashboard interface. At the top, there are two tabs: 'Dashboard < wordpress — WordPress' and 'Dashboard < wordpress — WordPress'. The right tab shows the user 'Howdy, wpuser'. The dashboard features a prominent 'Welcome to WordPress!' message with a 'Dismiss' button. Below this, there are three main sections: 'Author rich content with blocks and patterns', 'Customize your entire site with block themes', and 'Switch up your site's look & feel with Styles'. Each section includes a brief description and a link: 'Add a new page', 'Open site editor', and 'Edit styles' respectively. At the bottom, there are two widgets: 'Site Health Status' (showing 'No information yet...') and 'Quick Draft' (with fields for 'Title' and 'Content').

Not Secure — 138.197.122.185

Dashboard < wordpress — WordPress

Howdy, wpuser

Screen Options ▾ Help ▾

Dashboard

Welcome to WordPress!

[Learn more about the 6.8.1 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 blocks and patterns.

[Open site editor](#)

Switch up your site's look & feel with Styles

Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?

[Edit styles](#)

Site Health Status

No information yet...

Site health checks will automatically run periodically to gather information about your site. You can also [visit the Site Health screen](#) to gather information about your site now.

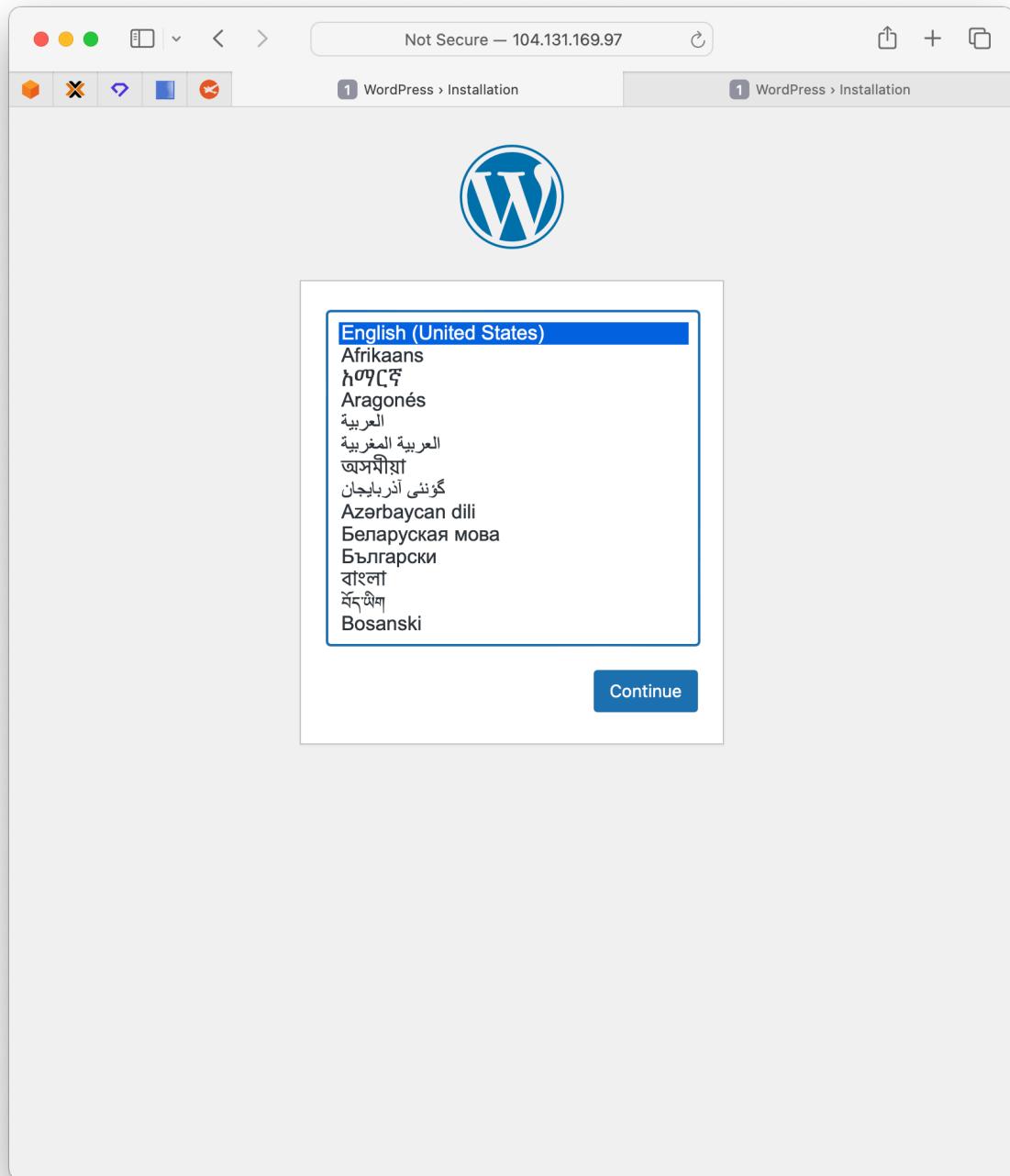
Quick Draft

Title

Content

What's on your mind?

- Use <http://104.131.169.97/wordpress/> to pull up WordPress for CoryS-2504-WP1



The screenshot shows the WordPress dashboard with a dark theme. At the top, a large banner says "Welcome to WordPress!" with a "Dismiss" button. Below the banner, there are three main sections: "Author rich content with blocks and patterns", "Customize your entire site with block themes", and "Switch up your site's look & feel with Styles". Each section has a description and a link to "Add a new page", "Open site editor", and "Edit styles" respectively. At the bottom, there are two widgets: "Site Health Status" (which says "No information yet...") and "Quick Draft" (with fields for "Title" and "Content").

Not Secure — 104.131.169.97

1 Dashboard < wordpress — WordPress

1 WordPress > Installation

Howdy, wpuser

Dashboard

X Dismiss

Welcome to WordPress!

Learn more about the 6.8.1 version.

Author rich content with blocks and patterns

Customize your entire site with block themes

Switch up your site's look & feel with Styles

Add a new page

Open site editor

Edit styles

Site Health Status

No information yet...

Site health checks will automatically run periodically to gather information about your site. You can also [visit the Site Health screen](#) to gather information about your site now.

Quick Draft

Title

Content

What's on your mind?

FINAL SCRIPT

Droplet1.yml

```
---
- name: Create Digital Ocean Droplets
  hosts: localhost
  connection: local
  gather_facts: false

  vars:
    droplet_name_prefix: "CoryS-2504-WP"
    region: "nyc3"
    size: "s-1vcpu-1gb"
    image: "ubuntu-22-04-x64"
    ssh_key_name: "Cory-Searcy-2504"
    number_of_droplets: 2
    droplets:
      - CoryS-2504-WP1
      - CoryS-2504-WP2

  tasks:
    - name: Create SSH Key in Digital Ocean
      digital_ocean_sshkey:
        name: "{{ ssh_key_name }}"
        ssh_pub_key: "{{ lookup('file', '~/.ssh/id_rsa.pub') }}"
        register: ssh_key_result

    - name: Create Droplets
      digital_ocean_droplet:
        name: "{{ droplet_name_prefix }}{{ item }}"
        region: "{{ region }}"
        size: "{{ size }}"
        image: "{{ image }}"
        ssh_keys: "{{ ssh_key_result.data.ssh_key.id }}"
        unique_name: yes
        state: present
      loop: "{{ range(1, number_of_droplets + 1) | list }}"
      register: droplet_result

    - name: Wait for droplets to be created
      pause:
        seconds: 30

    - name: Add new droplets to host group
      add_host:
        name: "{{ item.networks.v4[0].ip_address }}"
        groups: droplets
      loop: "{{ droplet_result }}"
      when: item.networks is defined and item.networks.v4 | length > 0
```

`lamp_wordpress.yml`

```
---
```

- `name`: Install LAMP and deploy WordPress on Ubuntu
 - `hosts`: webservers
 - `become`: `true`
 - `vars_files`:
 - `group_vars/all.yml`
- `tasks`:
 - `name`: Update apt cache
 - `apt`:
 - `update_cache`: `yes`
 - `name`: Install Apache, MySQL, PHP and dependencies
 - `apt`:
 - `name`:
 - `apache2`
 - `mysql-server`
 - `php`
 - `php-mysql`
 - `php-curl`
 - `php-gd`
 - `php-mbstring`
 - `php-xml`
 - `php-xmlrpc`
 - `php-soap`
 - `php-intl`
 - `unzip`
 - `curl`
 - `libapache2-mod-php`
 - `state`: `present`
 - `name`: Start and enable Apache
 - `service`:
 - `name`: `apache2`
 - `state`: `started`
 - `enabled`: `yes`
 - `name`: Start and enable MySQL
 - `service`:
 - `name`: `mysql`
 - `state`: `started`
 - `enabled`: `yes`
 - `name`: Ensure Python MySQL library is installed
 - `apt`:
 - `name`: `python3-pymysql`
 - `state`: `present`

```

- name: Set MySQL root password
  mysql_user:
    name: root
    host: localhost
    password: "{{ mysql_root_password }}"
    login_unix_socket: /var/run/mysqld/mysqld.sock
  ignore_errors: true

- name: Create WordPress database
  mysql_db:
    name: "{{ db_name }}"
    state: present
    login_user: root
    login_password: "{{ mysql_root_password }}"

- name: Create WordPress user
  mysql_user:
    name: "{{ db_user }}"
    password: "{{ db_password }}"
    priv: "{{ db_name }}.*:ALL"
    host: "%"
    state: present
    login_user: root
    login_password: "{{ mysql_root_password }}"

- name: Download WordPress
  get_url:
    url: "{{ wordpress_url }}"
    dest: /tmp/latest.tar.gz

- name: Extract WordPress
  unarchive:
    src: /tmp/latest.tar.gz
    dest: /var/www/html/
    remote_src: yes

- name: Set permissions for WordPress directory
  file:
    path: /var/www/html/wordpress
    state: directory
    recurse: yes
    owner: www-data
    group: www-data
    mode: '0755'

- name: Copy wp-config.php
  template:

```

```

src: wp-config.php.j2
dest: /var/www/html/wordpress/wp-config.php
owner: www-data
group: www-data
mode: '0644'

- name: Enable Apache rewrite module
  apache2_module:
    name: rewrite
    state: present

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

```

LESSONS LEARNED

One lesson I learned from this would have to be that even if everything is set up perfectly, you can still have connectivity issues.

APPENDIX A | DOCUMENT ERRORS & SOLUTIONS

1 - Error Title: Can't reach CoryS-2504-WP2

```

fatal: [CoryS-2504-WP2]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: root@167.172.30.107: Permission denied (publickey).", "unreachable": true}

```

Error explanation: Everything is set up correctly.

Solution: To fix this issue, I simply continued to try to run the playbook until it connected.

END