Writing Assignment #2

A Manual for Setting up a WordPress Website on the Local Linux Server with Vagrant

Yuji Shimojo

WRTG 393

Instructor: Claudia M. Caruana

November 21, 2016

Introduction

This is a manual for setting up a WordPress website on your local Linux server with Vagrant in 12 steps. This manual covers building a local Linux server, installing and setting up WordPress, and getting started your first WordPress post.

What is WordPress?

WordPress is a free and open-source content management system. You can easily create a blog system by using it. The official website of WordPress is https://wordpress.org/.

Previous knowledge

You need basic knowledge of UNIX commands, Linux, Apache, PHP, and MySQL.

Advance preparations

You need to install Cyberduck and Oracle VM VirtualBox into your computer.

Environment

I am showing my overall environment in which I created this manual as follows.

- OS X 10.11 El Capitan
- Cyberduck 5.2.2
- Vagrant 1.8.1
- Oracle VM VirtualBox 5.0.20
- CentOS 7.2.1511
- Apache 2.4.6

- PHP 5.6.21
- MySQL 5.7.16
- WordPress 4.6.1

Instructions

1. Create a virtual machine with Vagrant.

   I. Open Terminal.app.

   II. Add a Vagrant base box of 64bit CentOS 7.0. Vagrant base boxes are package formats for Vagrant environments.

   ```
   $ vagrant box add CentOS7.0 https://github.com/tommy-muehle/puppet-vagrant-boxes/releases/download/1.1.0/centos-7.0-x86_64.box
   ```

   III. Make directories for your virtual machine.

   ```
   $ mkdir ~/MyVagrant
   $ cd MyVagrant
   $ mkdir CentOS7.0
   $ cd CentOS7.0
   ```

   IV. Initiate Vagrant.

   ```
   $ vagrant init CentOS7.0
   ```

   V. Open Vagrantfile.

   ```
   $ vi Vagrantfile
   ```

   VI. Uncomment the following line.

   ```
   config.vm.network "private_network", ip: "192.168.33.10"
   ```

   VII. Run the virtual machine.

```
$ vagrant up
```

VIII. Log in to the server via SSH.

```
$ vagrant ssh
```

IX. Change locale setting of CentOS 7, which is German by default.

```
$ sudo localectl set-locale LANG=en_US.utf8
$ cat /etc/locale.conf
LANG=en_US.utf8
```

X. Change keymap setting of CentOS 7.

```
$ sudo localectl set-keymap us
$ cat /etc/vconsole.conf
KEYMAP=us
```

XI. Reflect the locale settings.

```
$ source /etc/locale.conf
$ localectl status
   System Locale: LANG=en_US.utf8
       VC Keymap: us
      X11 Layout: us
       X11 Model: pc105+inet
     X11 Options: terminate:ctrl_alt_bksp
```

2. Install and configure Apache.

I. Install Apache.

```
$ sudo yum -y install httpd
```

II. Start Apache process and make it always running.

```
$ sudo systemctl start httpd.service
$ sudo systemctl enable httpd.service
```

III. Stop firewalld and make it always stop.

```
$ sudo systemctl stop firewalld
$ sudo systemctl disable firewalld
```

IV. Open the configure file.

```
$ sudo vi /etc/httpd/conf/httpd.conf
```

V. Uncomment the following line if it is commented out, or add the line if it is not existing.

VI. Add the following lines to enable use of .htaccess

```
<Directory "/var/www/html/wordpress">
    Options FollowSymLinks
    AllowOverride All
    Order Allow,Deny
    Allow from all
</Directory>
```

3. Install PHP.

I. Add Remi repository.

```
$ sudo yum -y install epel-release
$ cd /etc/yum.repos.d
$ sudo wget http://rpms.famillecollet.com/enterprise/remi.repo
```

II. Install PHP 5.6 from Remi repository.

```
$ sudo yum -y install --enablerepo=remi --enablerepo=remi-php56 php php-opcache php-devel php-pear php-mbstring php-mcrypt php-mysqlnd
```

III. Confirm the version of PHP.

```
$ php -v
PHP 5.6.21 (cli) (built: Apr 28 2016 07:39:37)
```

4. Install MySQL.

IV. Add yum repository of MySQL 5.7.

```
$ sudo yum localinstall http://dev.mysql.com/get/mysql57-community-release-el7-7.noarch.rpm
```

V. Install MySQL Community Server.

```
$ sudo yum -y install mysql-community-server
```

VI. Confirm the version of MySQL.

```
$ mysqld --version
mysqld  Ver 5.7.16 for Linux on x86_64 (MySQL Community Server (GPL))
```

VII. Make MySQL always running.

```
$ sudo systemctl enable mysqld.service
```

5. Create a table in MySQL database.

I. Log in to MySQL as root user.

```
$ mysql -u root -p
```

II. Create a table for WordPress with a user name and a password.

```
mysql> create database example_wordpress;
mysql> grant all on example_wordpress.* to dbuser@localhost identified by 'your password';
```

6. Modify your Hosts file.

    I. Open your Hosts file via Terminal.app.

    ```
    $ sudo vi /private/etc/hosts
    ```

    II. Add a mapping between dev.example.com and 192.168.33.10 in your Hosts file.

    ```
    192.168.33.10   dev.example.com.
    ```

7. Download wordpress and deploy it into the server.

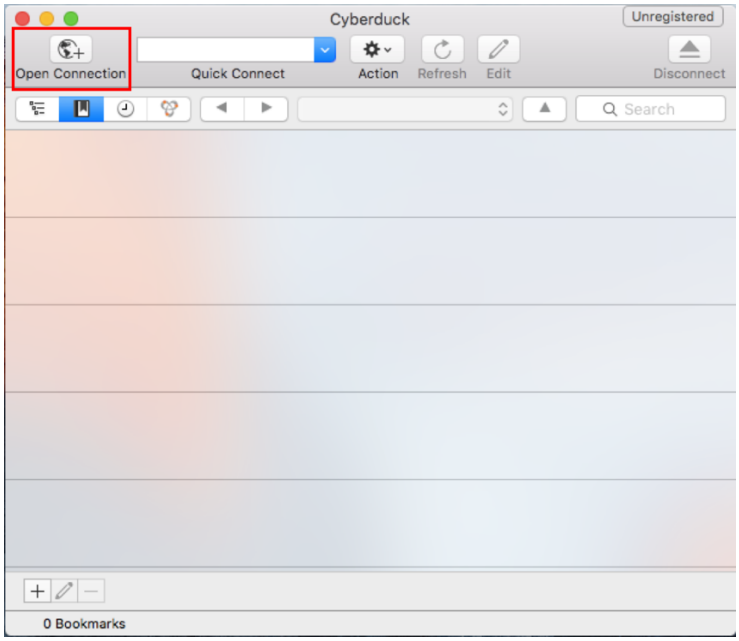
    I. Download the latest version of WordPress as a zip file to your local computer from the WordPress official website.

    II. Unzip the .zip file.

    III. Open Cyberduck.app.

    IV. Click **Open Connection** in the upper left hand corner.

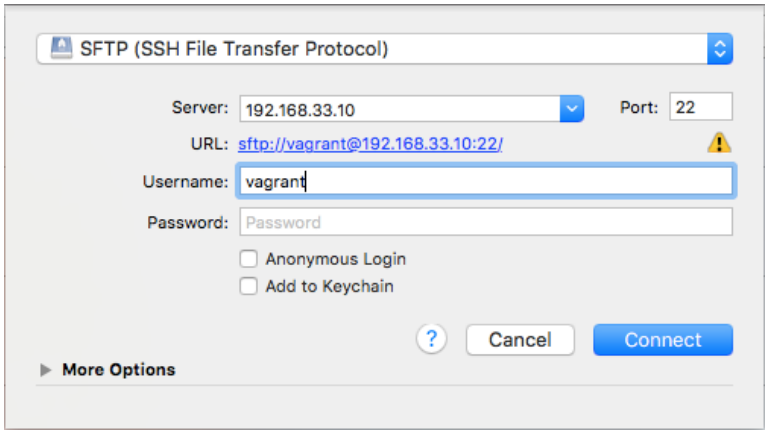
    Figure 1



V. Select **SFTP** as the network protocol from the drop-down list, input **192.168.33.10** into the **Server** text field, enter **Username** as shown in Figure 2. Then click the **Connect** button.

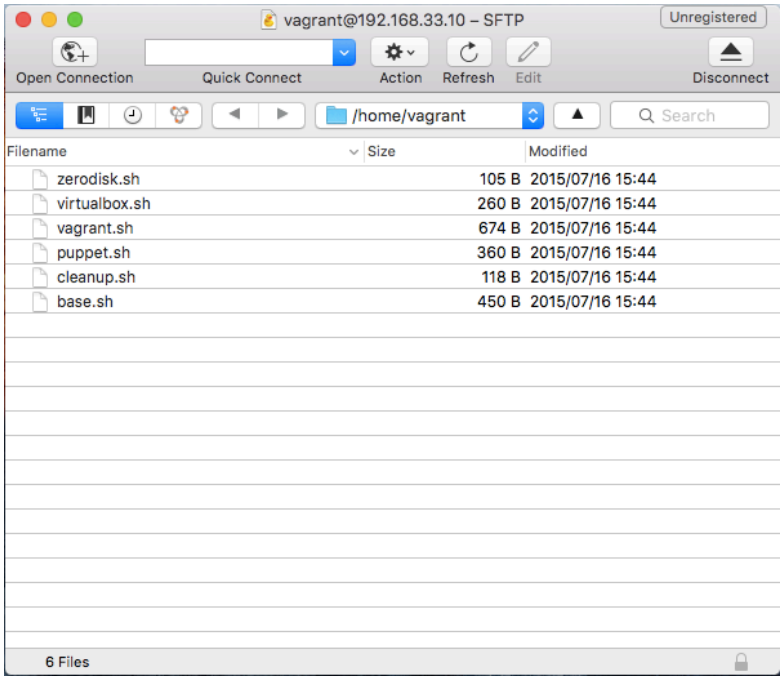
* *you don't need to enter Password when you log in as **vagrant** user.*

Figure 2



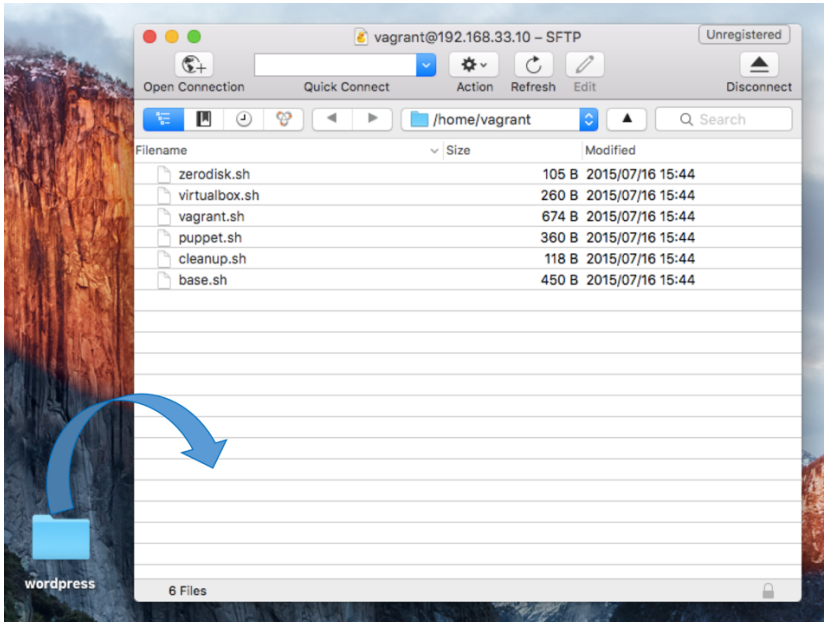
VI. If you have successfully logged in to the server, you will see the home directory of vagrant user in the virtual machine as shown in Figure 3.

Figure 3



VII. Drag the wordpress folder you unzipped and drag into the home directory as shown in Figure 4.

Figure 4



VIII. Open Terminal.app and log in to the server via SSH again.

IX. Move the wordpress directory in your home directory into /var/www/html, which is known as the document root.

X. $ sudo mv wordpress /var/www/html/

XI. Change the owner of wordpress to apache user.

$ cd /var/www/html

$ sudo chown -R apache:apache wordpress

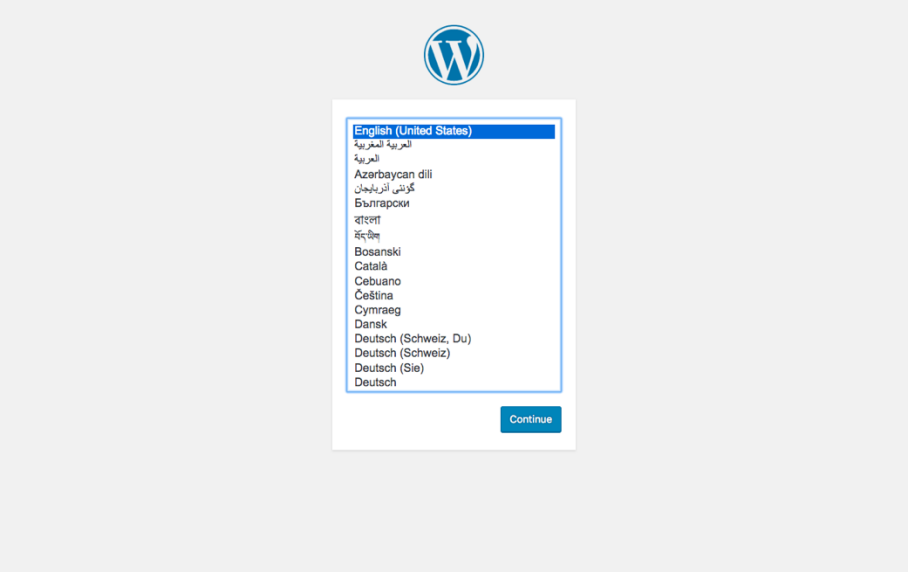
8. Install WordPress via WordPress dashboard.

I. Launch a web browser.

II. Type http://dev.example.com/wordpress in the address bar and press the **Enter** key.

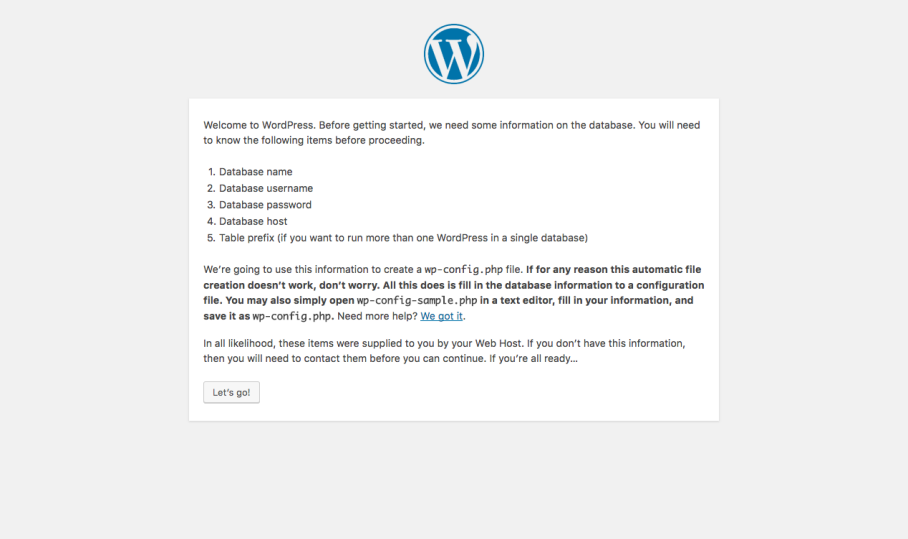
III. If you have successfully accessed WordPress installing page, you will see the page as shown in Figure 5. Then select your language and Click the **Continue** button.

Figure 5



IV. Click the **Let's go** button.

Figure 6

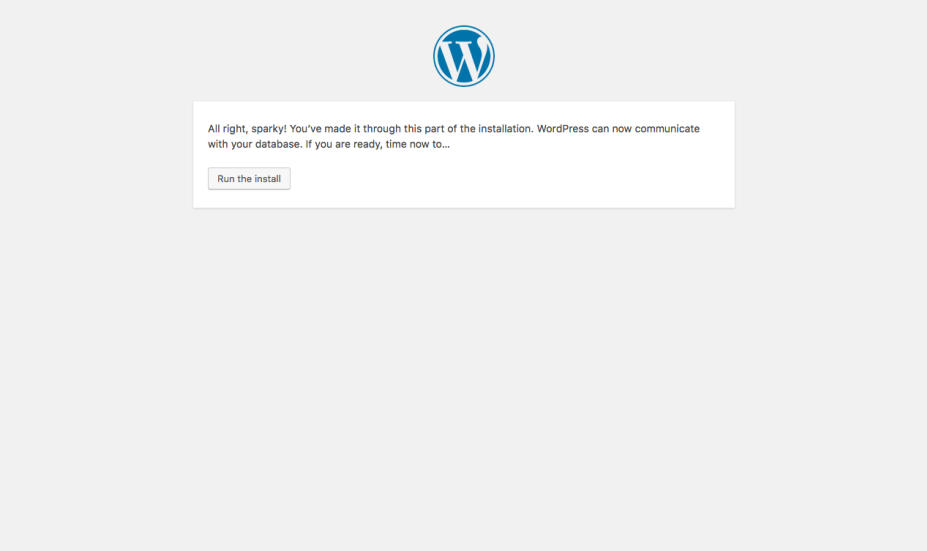


V. Type **example_wordpress** in the **Database Name**, **dbuser** in the **Username**, and **your password** you set when you created the database table for WordPress. Then click the **Submit** button.

Figure 7

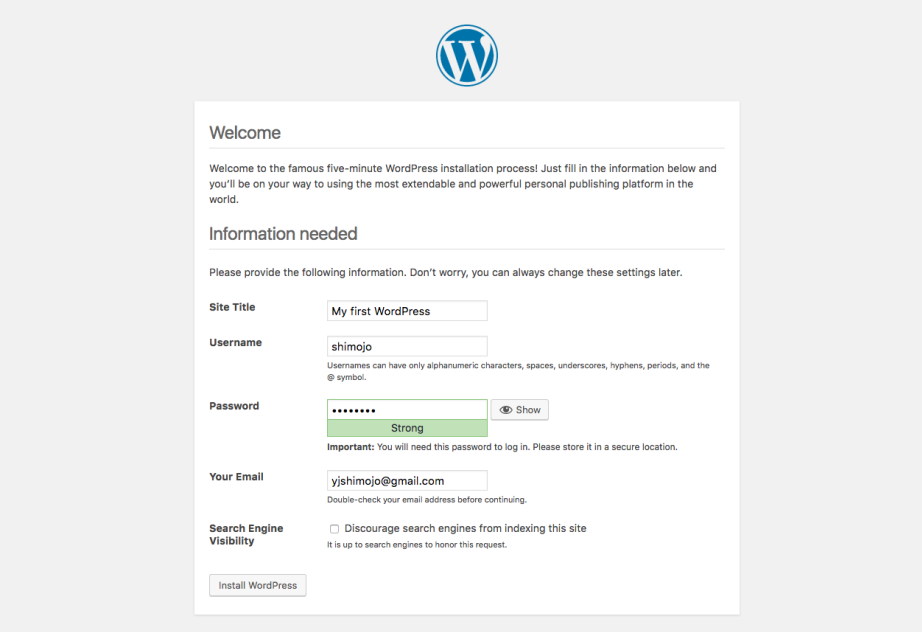
VI. Click the **Run** the install button.

Figure 8



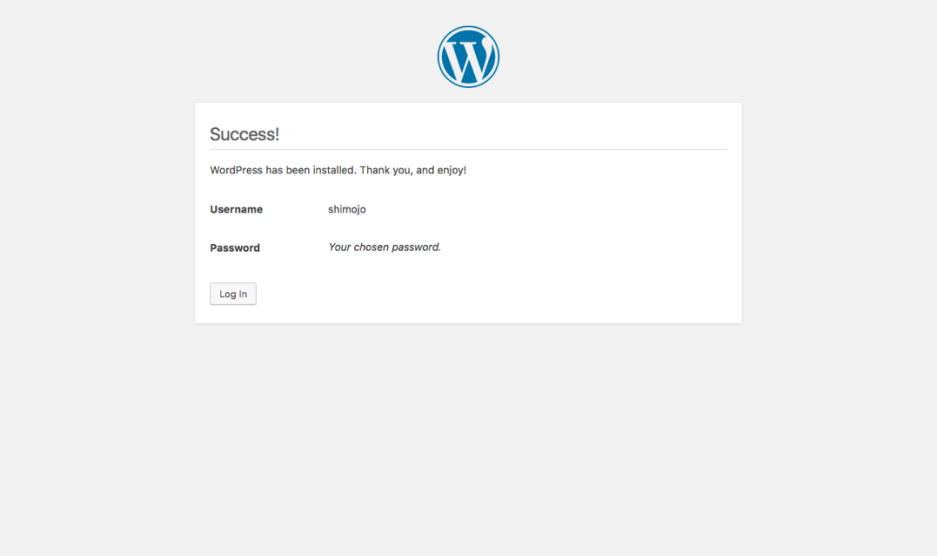
VII. Set your arbitrary **Site Title**, **Username**, **Password**, **Your Email**. Uncheck the **Search Engine Visibility** box. Then click the **Install WordPress** button.

Figure 9



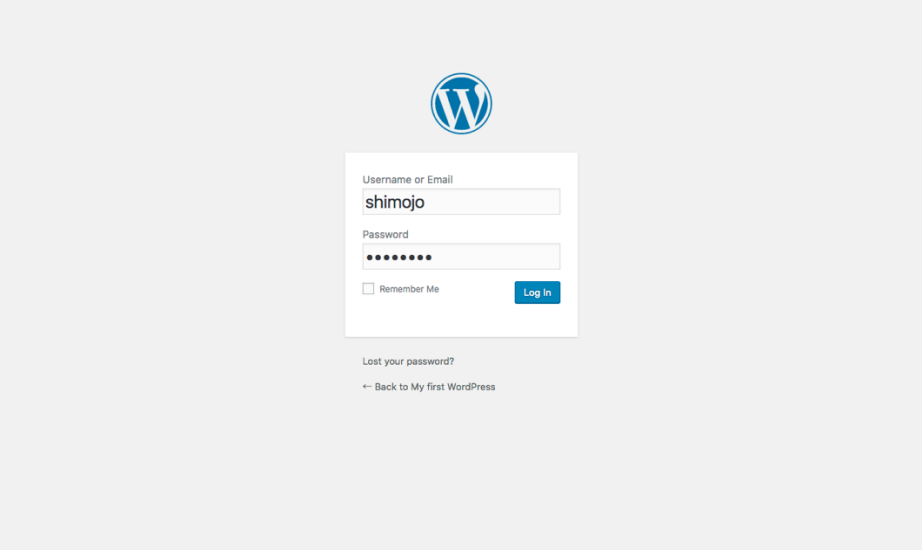
VIII. You have successfully installed WordPress. Click the **Log in** button.

Figure 10



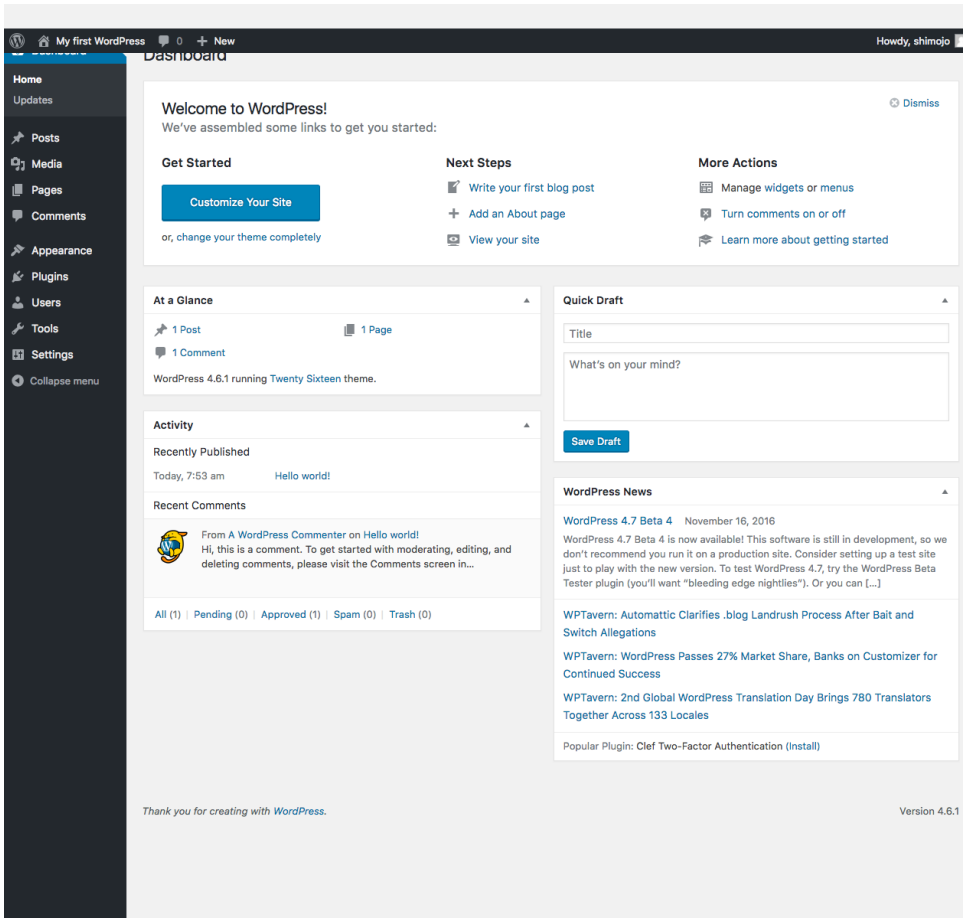
IX. Enter your **username or email** and **password** you set when you installed WordPress. Then click the **Log in** button.

Figure 11



X. If you have successfully logged in to WordPress, you will see the page as shown in Figure 12, which is called dashboard.

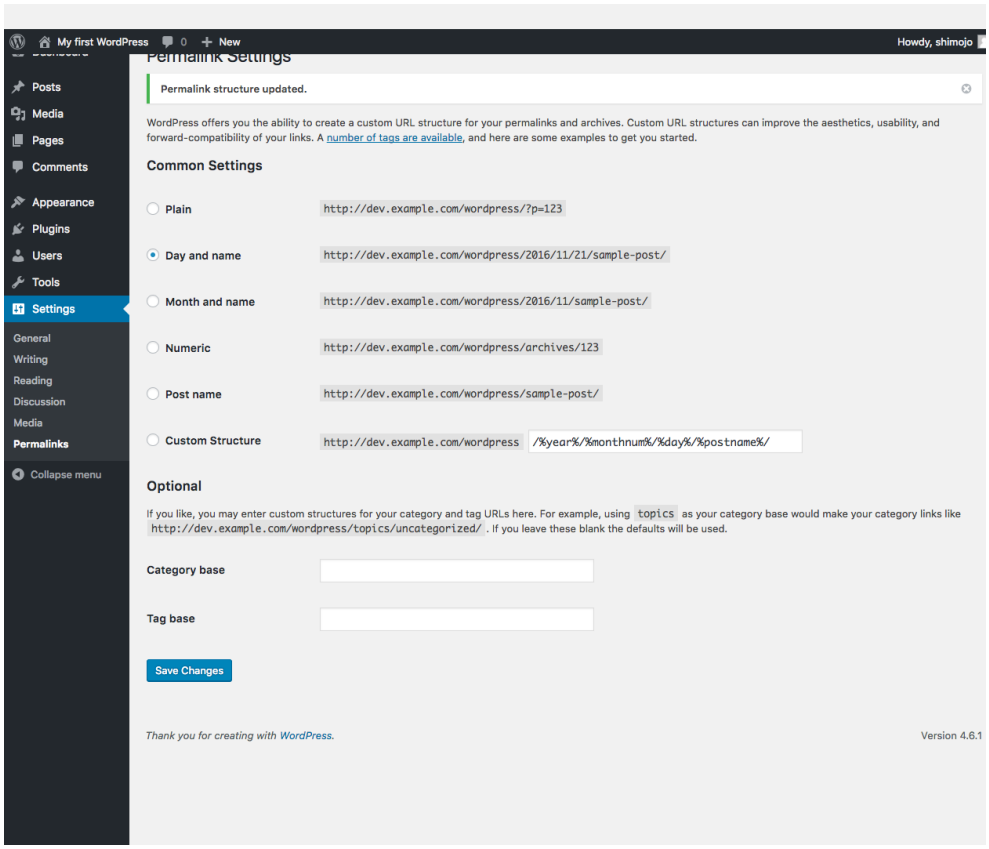
Figure 12



9. Change permalink structure

   I. Select **Settings** in the left side bar and click **Permalinks** in the **Settings** menu.

Figure 13



II. Change permalink structure **default** to **Day and name**.

* *The short string of characters "sample-post" is called Slug which represents each your article.*

III. Click the **Save Changes** button.

* *At this time, a .htaccess file will automatically be created in the wordpress home directory in the server. The .htaccess file allows to rewrite URL requests. Note that if you didn't enable URL rewriting when you configure the httpd.conf unsuccessfully, you will get a 404 Page Not Found Error when you access your post after this.*

10. Modify your Hosts file.

I. Open Terminal.app and log in to the server via SSH again.

```
$ cd Documents/MyVagrant/CentOS7.0/
```

```
$ vagrant ssh
```

II. Create 'uploads' directory under /var/www/html/wordpress/wp-content/.

```
$ cd /var/www/html/wordpress/wp-content
```

```
$ sudo mkdir uploads
```

III. Change the owner of 'uploads' directory.

```
$ sudo chown -R apache:apache uploads
```

IV. Give the full access permission to 'uploads' directory.

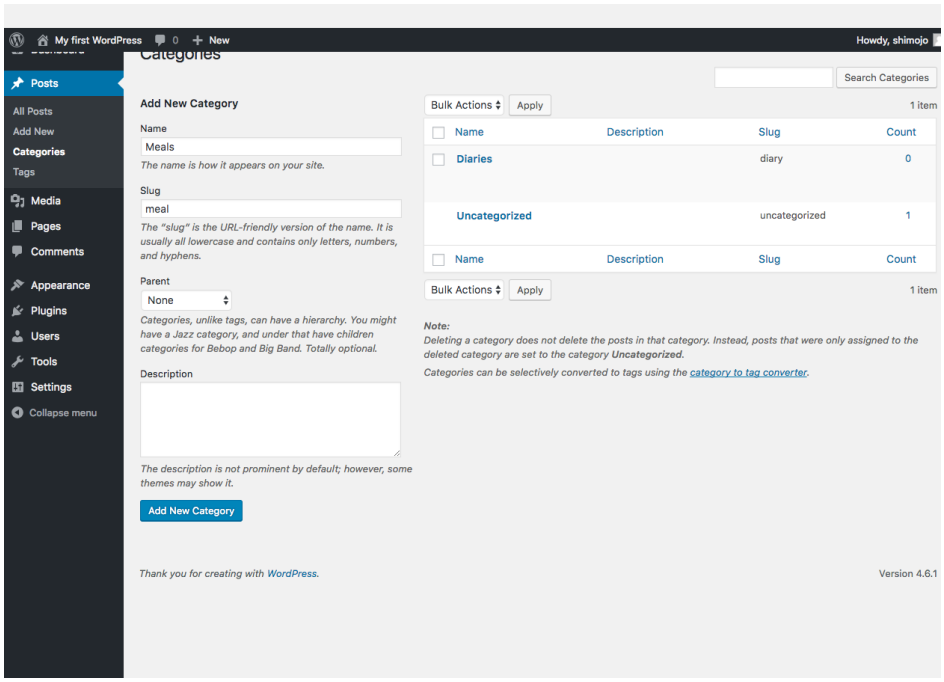
```
$ sudo chmod 777 uploads
```

11. Add new category

I. Go back to the dashboard.

II. Select **Posts** in the left side bar and click **Categories** in the **Posts** menu.

Figure 14

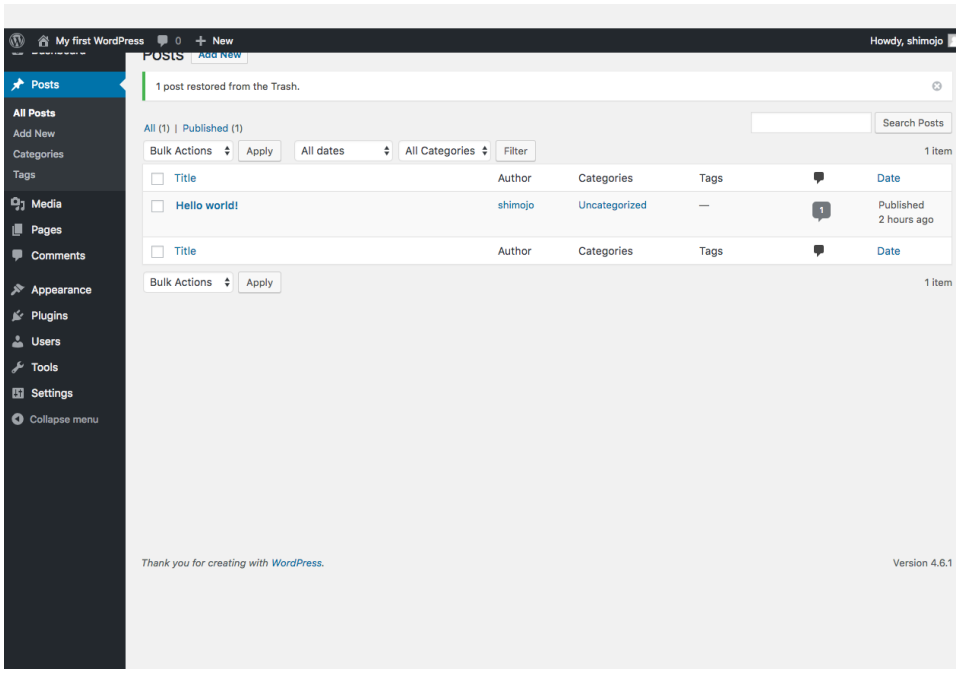


III. Decide **Name** and **Slug**. Then Click the **Add New Category** button.

12. Add new post

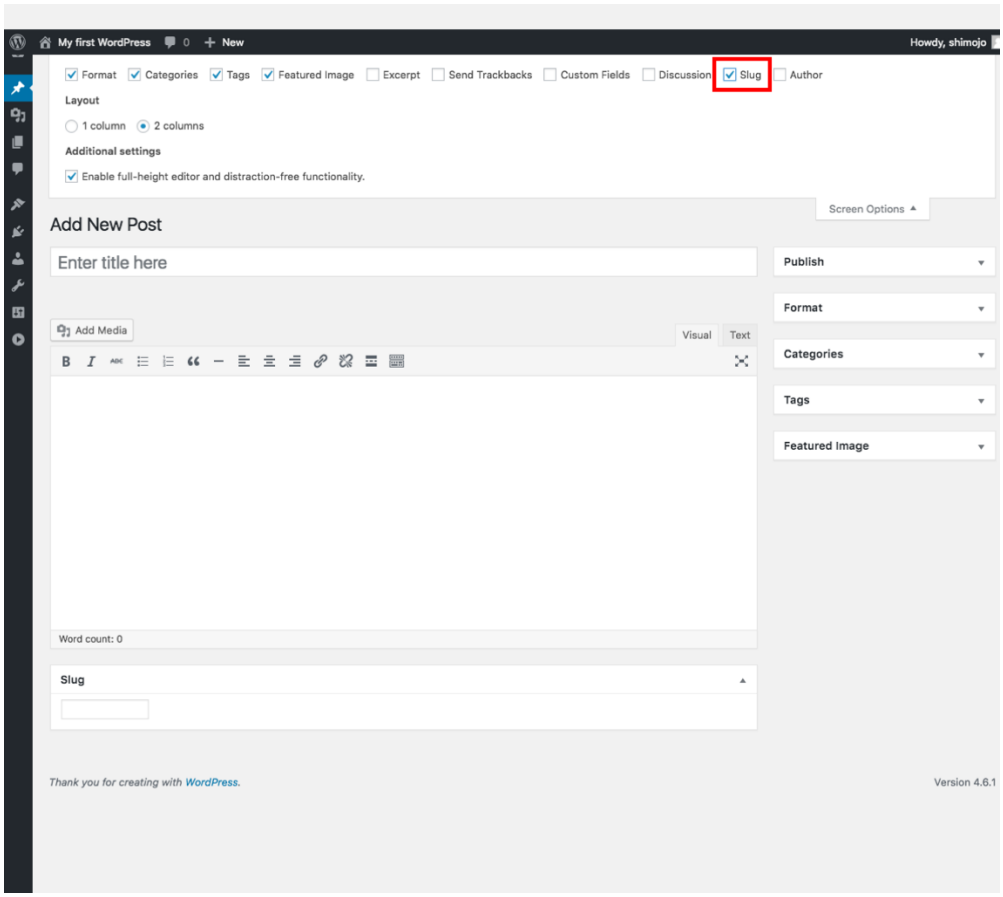
I. Select **Add New** in the Posts menu.

Figure 15



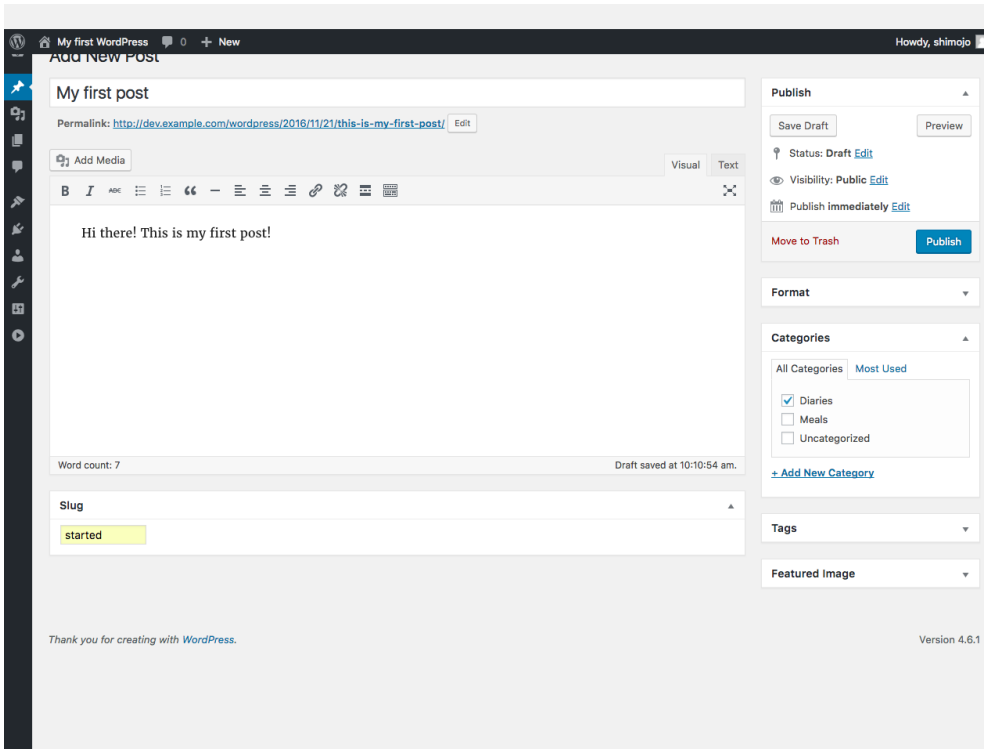
II. Before you add a new post, check the **Slug** box in the **Screen Options**.

Figure 16



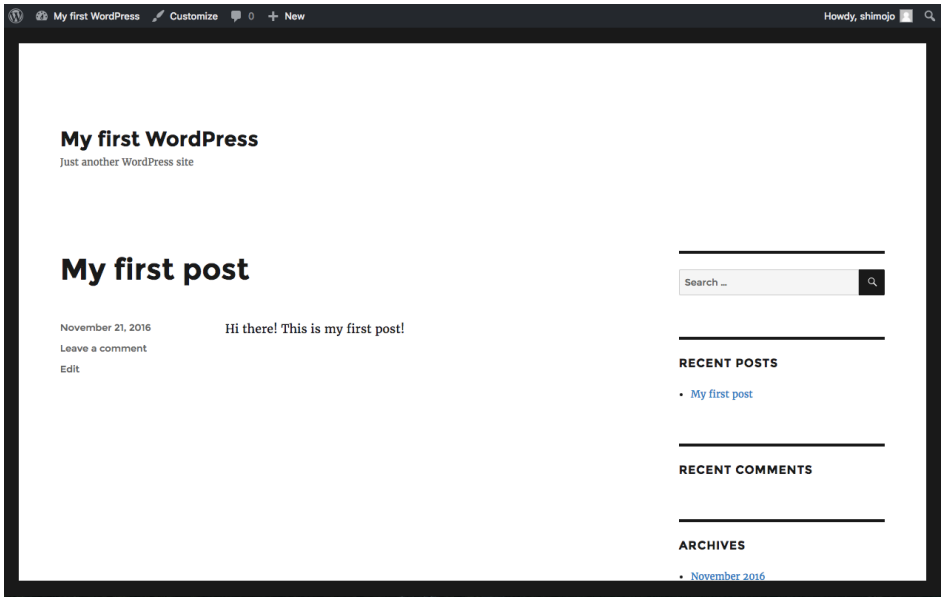
III. Enter a title, a main text, and a slug, and choose categories. Then click the **Publish** button in the right side side menu.

Figure 17



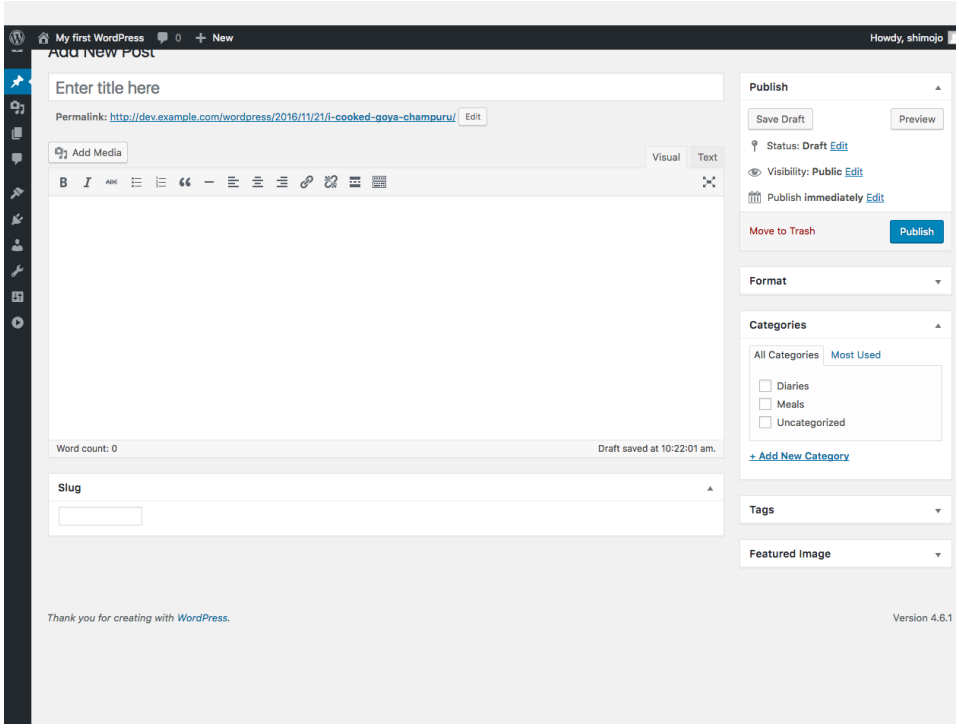
IV. Go to dev.example.com/wordpress. If you have successfully created a post, you will see the page as shown in Figure 18.

Figure 18



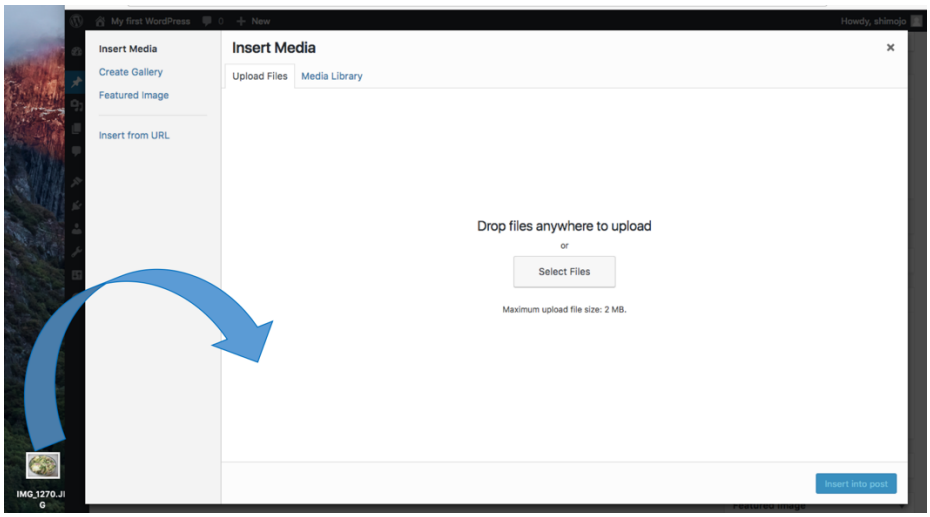
V. When you create a new post with an image, click the **Add Media** button.

Figure 19



VI. Insert your image by dragging it and dropping into the web browser to upload as shown in Figure 20.

Figure 20



VII. Enter a title, a main text, and a slug, and choose categories. Then click the **Publish** button in the right side side menu.

Figure 21

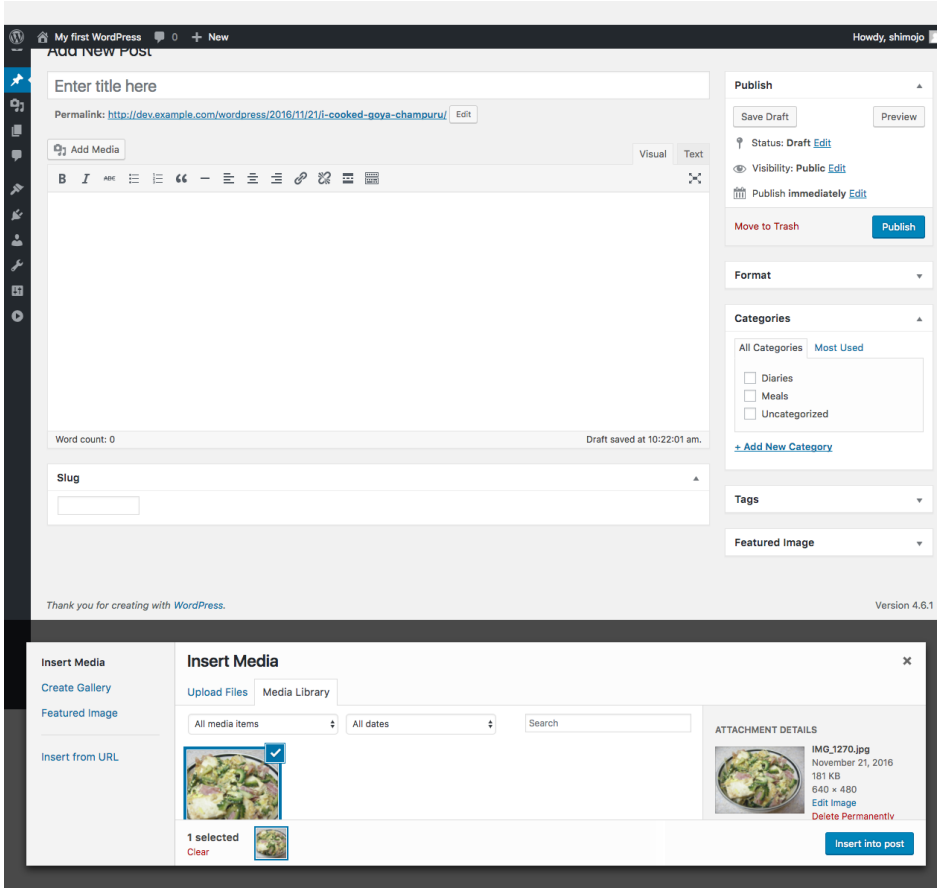
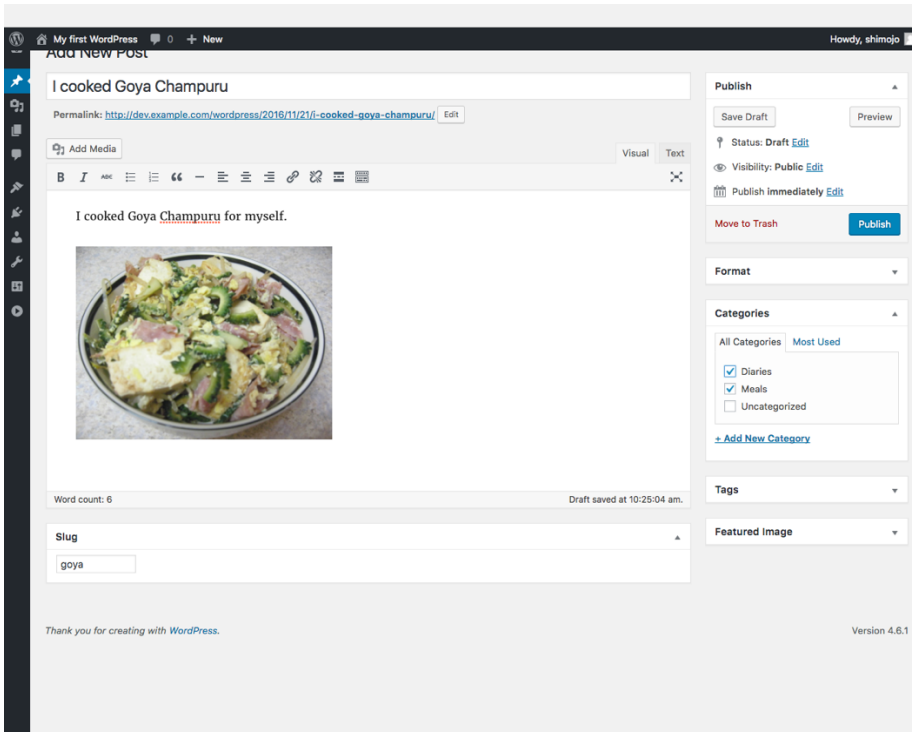
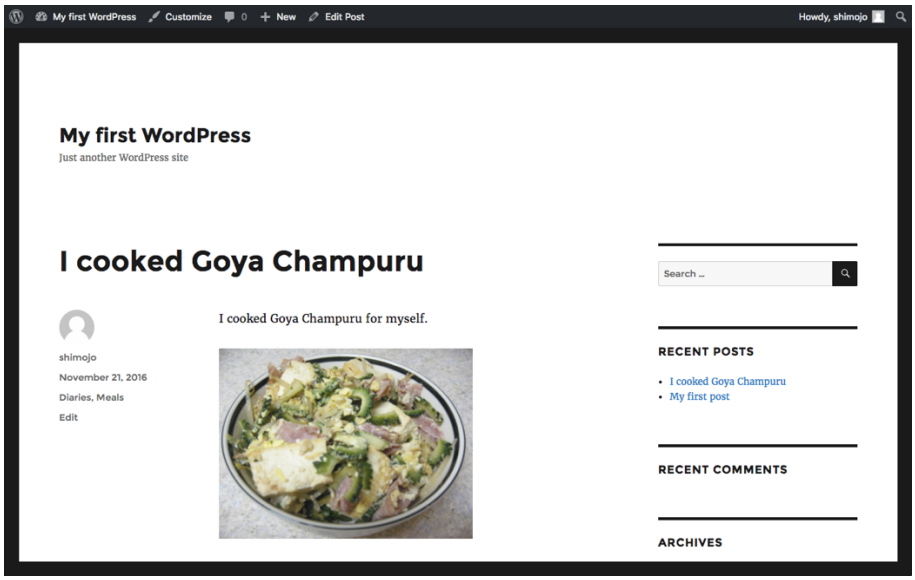


Figure 22



VIII. Access the link of the post such as http://dev.example.com/wordpress/2016/11/21/goya/. If you have successfully created a post, you will see the page as shown in Figure 23.

Figure 23



References

- HashiCorp, Inc. (n.d.). VAGRANT GETTING STARTED. Retrieved from https://www.vagrantup.com/docs/getting-started/
- Oracle Corporation and/or its affiliates. (n.d.). MySQL Documentation. Retrieved from http://dev.mysql.com/doc/
- The Apache Software Foundation. (n.d.). Apache HTTP Server Version 2.4 Documentation. Retrieved from https://httpd.apache.org/docs/2.4/
- The PHP Group. (n.d.). PHP Documentation. Retrieved from http://php.net/docs.php
- Wikimedia Foundation. (n.d.). The Developer's Guide to WordPress. Retrieved from https://en.wikibooks.org/wiki/The_Developer's_Guide_to_WordPress
- WordPress.org. (n.d.). WordPress Codex. Retrieved from https://codex.wordpress.org/