# LAMP stack/WordPress project SOP

*__NOTE__: Spin up 2 Amazon Linux EC2 instances (**web server** and **database server**).
**WebServer EC2:** please open ports 22 and 80.
**DataBase EC2:** please open port 3306 to the WebServer IP address and open port 22 to the world

**Apache -** The Apache web server is currently the most popular web server in the world, which makes it a great default choice for hosting a website. As a DevOps / SRE engineer your responsibility to make sure that **infrastructure is up and running**. You don't need to develop or know the content of the website.

## WebServer EC2:

1. Install apache - `yum install httpd`
2. Start apache - `systemctl start httpd.service`
3. Go to chrome and verify if it's working - `http://<your_server_IP_address>/` ***NOTE:** Please replace your_server_IP_address with your ec2 public IP address.
4. Enable apache to start on boot - `systemctl enable httpd.service`

**MySQL -** MariaDB is MySQL drop-in replacement. MariaDB is a community-developed fork of the MySQL relational database management system. In a nutshell, MariaDB works exactly like MySQL, hence we are going to use MariaDB in our project

## DataBase EC2:

1. Install MySQL (MariaDB) - `yum install mariadb-server mariadb`
2. Start MariaDB - `systemctl start mariadb`
3. Setup password for root user to be used when switching to mysql console: `mysql_secure_installation` It will prompt you to create password. Remember it.
4. Enable DB to start on boot - `systemctl enable mariadb.service`

**PHP** is the component of our setup that will process code to display dynamic content. It can run scripts, connect to our MySQL databases to get information, and hand the processed content over to our web server to display

## WebServer EC2:

1. Install epel repo - `amazon-linux-extras install`
2. Enable epel repo - `amazon-linux-extras enable epel`
3. Enable php74 repo - `amazon-linux-extras enable php7.4`
4. Clean yum metadata - `yum clean metadata`
5. Install required packages: `yum install mysql python-pip gcc mysql-devel php php-mysqlnd php-mbstring`
6. Install python MySqldb module (required to login to DB) - `pip install MySQL-python`

**Wordpress** is a free and open source website and blogging tool that uses PHP and MySQL. WordPress is currently the most popular CMS (Content Management System) on the Internet, and has over 20,000 plugins to extend its functionality. This makes WordPress a great choice for getting a website up and running quickly and easily

## Go to your DataBase EC2 server and create a database.

1. `mysql -u root -p` and enter password you created in DataBase EC2 Step 3.
2. `CREATE DATABASE wordpress;`
3. `CREATE USER 'wp_user'@'public ip of your webserver' IDENTIFIED BY 'your custom password';`
4. `GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'public ip of your webserver' IDENTIFIED BY 'your custom password';`
5. `FLUSH PRIVILEGES;`
6. `SELECT User, Host, Password FROM mysql.user;` - command to check if user created.
7. `exit;`

***NOTE:** Please write down or memorize DataBase name, User name, DataBase Password - you will be using them in next step.

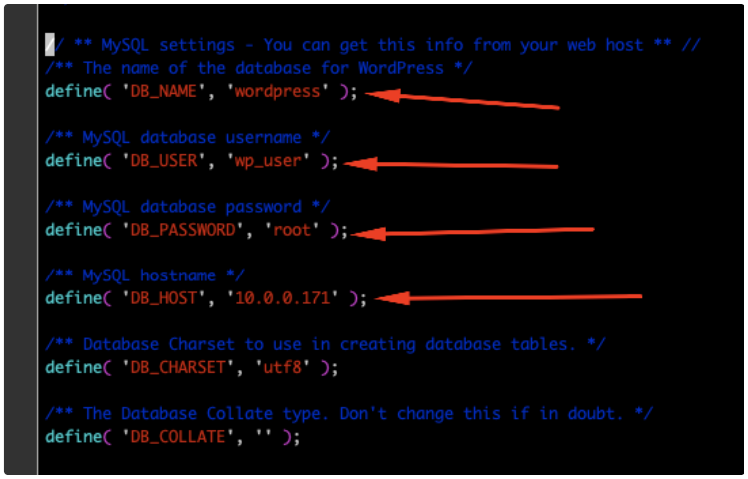
## Go to your webserver EC2 and install WordPress

1. `cd /tmp`
2. `wget http://wordpress.org/latest.tar.gz`
3. `tar xzvf latest.tar.gz`
4. `rsync -avP wordpress/ /var/www/html/`
5. `mkdir -p /var/www/html/wp-content/uploads`
6. `chown -R apache:apache /var/www/html/`
7. `cd /var/www/html`
8. `cp wp-config-sample.php wp-config.php`
9. `vim wp-config.php`

The only modifications we need to make to this file (**wp-config.php**) are to the parameters that hold our database information. We will need to find the section titled MySQL settings and change the **DB_NAME**, **DB_USER**, and **DB_PASSWORD** variables in order for WordPress to correctly connect and authenticate to the database that we created.

> ***NOTE**: after each configuration change apache needs to be **restarted***

See the example below



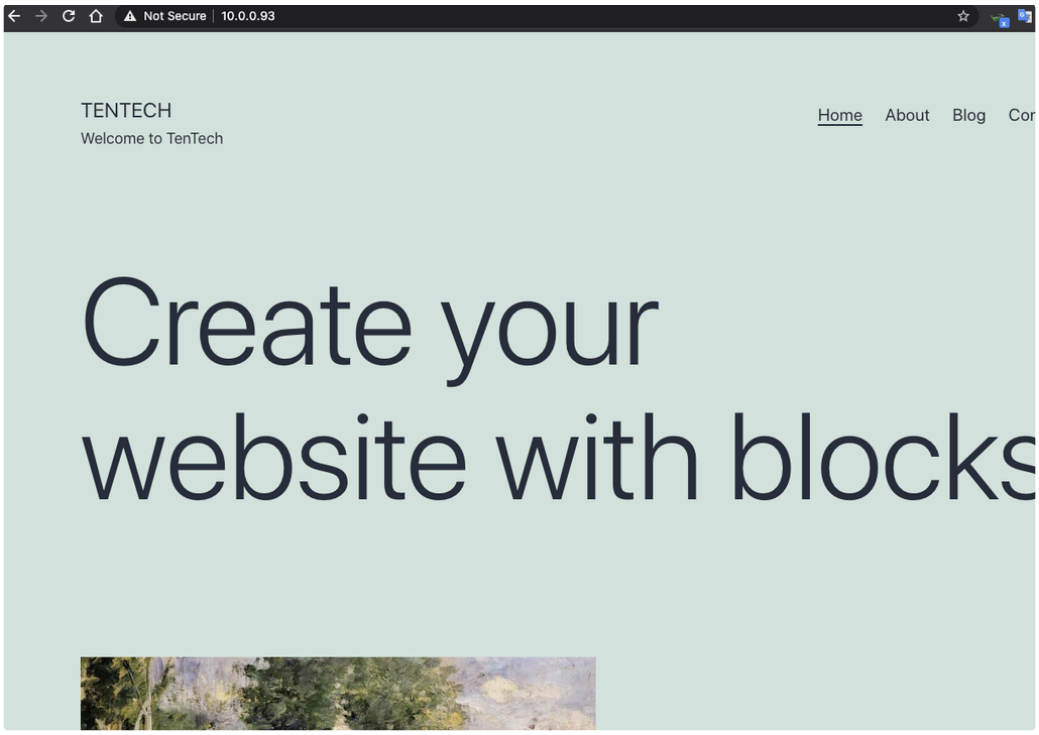

10. Once completed restart **httpd** process.
11. Complete the installation through the **Web Interface -** `http://<server_domain_name_or_IP`

Now that you have your files in place and your software is configured, you can complete the WordPress installation through the web interface. In your web browser, navigate to your server's domain name or public IP address:

`http://server_domain_name_or_IP`

First, you will need to select the language that you would like to install WordPress with. After selecting a language and clicking on **Continue**, you will be presented with the WordPress initial configuration page, where you will create an initial administrator account:


Once you are done put your webserver's ip into your browser and you should see your website



👍

👏

🎉

Be the first to add a reaction