WORDPRESS ON EC2

# Brief Summary of the Architecture

VPC (dev)

* Public subnet (dev public)
* internet gateway(dev-igw)
* public route table(dev-public-rt), subnet association to the public subnet(dev public)

EC2 (Wordpress-server)

* Amazon Linux 2
* attached to the dev VPC
* created an ssh key (EC2Tutorial.pem)
* attached to the public subnet(dev-public)
* Security group(wordpress-sg), allows http(80),ssh(22),sql(3306)
* EBS Volume (8gib)

# Summary of the configuration of wordpress-server. (Followed the AWS Documentation Mostly)

Connecting to the EC2 Server via SSH

* First locate the EC2Tutorial.pem key in cli
* Set permissions to this key to read only using the chmod 400 <ssh key.pem>
* Use ssh command;  **ssh -i "<ssh key.pem>"** [**ec2-user@ec2<-public-ip-address-of-your-instance>.compute-1.amazonaws.com**](mailto:ec2-user@ec2%3c-public-ip-address-of-your-instance%3e.compute-1.amazonaws.com)to login into your you EC2 machine
* After login your user might look similar to this: [ec2-user ~]$
* Run this command to make sure you have the latest software updates on your instace: **sudo yum update -y** (***Note to self:*** Break this down in more detail and explain what this command mean)

Installing Apache server and MariaDb database

* If your used the Amazon Linux 2 AMI (this command won’t run on Amazon Linux AMI, it’s only for v2) then you should also make sure you run this so that you have the latest mariadb version and php when we get to install the them, the command is: **sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2** (***Note to self:*** Break this down in more detail and explain what this command mean)
* Then install both the APACHE web server and the maridb server with this command: **sudo yum install -y httpd mariadb-server** (***Note to self:*** Break this down in more detail and explain what this command mean)
* Start the apache web server: **sudo systemctl start httpd**
* If you want to test to see if the apache web server can accessed via the internet, copy the public ip address from the instance and paste it in the browser, you should atleast see a test page for apache. **-> If you don’t see it make sure the security group you attached to the ec2 allows port 80(http)**

Add ec2-user(which is our server’s name) to the APACHE Document root group

* We add ec2-user to this group so that we have an ability to manipulate files in this directory, so we modify the ownership and permissions of the directory you add ec2-user to the apache group, to give the apache group ownership of the /var/www directory and assign write permissions to the group.
* So add the user(ec2-user) to the apache group with this command: **sudo usermod -a -G apache ec2-user** (***Note to self:*** Break this down in more detail and explain what this command mean)
* Change the group ownership of /var/www and its contents to the apache group: **sudo chown -R ec2-user:apache /var/www** (***Note to self:*** Break this down in more detail and explain what this command mean)
* To add group write permissions and to set the group ID on future subdirectories, so that we can edit and add to the group, use this command: **sudo chmod 2775 /var/www && find /var/www -type d -exec sudo chmod 2775 {} \;** (***Note to self:*** Break this down in more detail and explain what this command mean)
* To be able to edit the files in the /var/www group, you again add write permissions to it with this command: **find /var/www -type f -exec sudo chmod 0664 {} \;** (***Note to self:*** Break this down in more detail and explain what this command mean)
* Now, ec2-user (and any future members of the apache group) can add, delete, and edit files in the Apache document root, enabling you to add content, such as a static website or a PHP application.

Securing the MariaDB database

* Start the MariaDB server: **sudo systemctl start mariadb**
* Then run: **sudo mysql\_secure\_installation**
* When you press **enter** you’ll prompted for a password but since you haven’t set any password, there is no default password, press **Enter** again
* You’ll be asked if you want to set a password type **Y** and then type a secured password.
* After you set your password the rest of the prompts you shout type **Y.**

Install WordPress

* Using the wget command you download and install the latest Wordpress installation package: **wget** [**https://wordpress.org/latest.tar.gz**](https://wordpress.org/latest.tar.gz)
* Unzip the installation package file with: **tar -xzf latest.tar.gz**

Create a Database for the WordPress.

* Because word press needs to save things like comments, blog posts, and other things, we need to create a database and a user that authorizes to read and save information to it.
* First start the Mariadb database we created earlier: **sudo systemctl start mariadb**
* Login as the databse root user using this command: **mysql -u root -p.** This command also prompt you for a password(which we created earlier when securing the database)
* When you’re logged in change to the mysql db using: **use mysql** command.
* Create a user and a password for your MYSQL database using this command: **CREATE USER 'wordpress-user'@'localhost' IDENTIFIED BY 'your\_strong\_password';**
* Create the database and give it a more descriptive name using this command: **CREATE DATABASE `wordpress-db`; Note that your database name is inside the backticks, which are usually located above the Tab key on your keyboard.**
* Grant full privileges for the databse to the wordpress user we created earlier**: GRANT ALL PRIVILEGES ON `wordpress-db`.\* TO "wordpress-user"@"localhost";**
* Run the **FLUSH PRIVILEGES** command so that the privileges we set earlier are updated.