R.Mohan Rajan
CEG, Anna University
30 November 2015

Installing Laravel on AWS Server

(or any cloud servers , in that case.)

Laravel 5.1 is a PHP Framework and has few requirements. PHP >= 5.5.9, OpenSSL PHP Extension, PDO PHP Extension, Mbstring PHP Extension, Tokenizer PHP Extension.

To know how to install laravel on was server , we go by the following ways:

1.Creating an AWS Instance (or any server instance on a cloud server)

2.Installing the basic NGINX , MYSQL , PHP ,MYCRYPT on the created instance.

3.Initialize git core and start to clone the website contents into the server.

4.Install composer onto the server

5.Change the .htaccess files

1.Creating an AWS Instance

1. Sign up for AWS

Visit http://aws.amazon.com/console and choose Sign Up. Have a credit card and a phone ready to verify your identity and to add payment information.

Even though you're entering your credit information, what we're setting up today will keep you on the Free tier, so you don't have to worry about being charged immediately.

AWS Free Tier

2. Visit the AWS console

Once you're signed up and have verified your account, visit the AWS console. Click on the "EC2" button to take you to the management console for EC2, or "Elastic Compute Cloud"--Amazon's service for creating and managing Virtual Machines.

3. Launch a new Instance

From here, click "Launch Instance".

AWS ECS Dashboard

4. Choose your Machine Image

This allows you to specify which Machine Image--that is, which pre-created recipe for a Virtual Machine--you'd like to base this instance off of. AWS Select Machine image

5. Choose Instance Type

For this demo, we'll go for the lowest power, free option: t2.micro.

AWS Instance Type

Rather than Launching now, let's walk piece by piece through the configuration process.

6. Configure Instance Details

You can now configure all the specific configuration details for this instance.

The defaults here are fine for a demo, although if you plan to rely on this simple server for anything you'll probably want to check Enable Termination Protection so the server will reboot if anything happens to shut it down. When you're done, move to the next screen.

AWS Instance Details

7. Add Storage

We can configure the amount and type of storage our instance will have available. The default is an 8GiB SSD drive, so let's just keep that as-is and move on.

AWS Add Storage

8. Tag Instance

AWS allows you to tag each instance with up to 10 key/value pairs. This can be useful if you want to sort or add permissions to instances later (using IAM roles) based on client (Client=Bob), environment (Environment=Staging), management service (Managed-By=Forge), or more.

If this is confusing, feel free to just skip it. I just added Managed-By=Forge.

AWS Tag Instance

9. Configure Security Group

Security Groups allow you to associate multiple instances together with a single set of security permissions. Security Groups allow you to both apply the same settings to multiple instances, and create an instant firewall surrounding just the members of that group--one of the primary permissions options for rules is "only members of this security group."

Therefore, you'll want to create a new security group for each project you're working on.

Depending on your needs, you'll want to add an entry for each. I added SSH, HTTP, MySQL, and you could also add IMAP/SMTP/POP3 if you need mail. You'll see the dropdown contains many other options for adding security access rules.

AWS Security Groups

For now, add:

An SSH entry with "Anywhere" (you'll want to lock this down to just specific IP addresses later)

An HTTP entry with "Anywhere"

A MySQL entry for "My IP""

SECURITY CAUTION FROM AMAZON: If you use 0.0.0.0/0 ("Anywhere") for SSH, you enable all IP addresses to access your instance using SSH. This is acceptable for a short time in a test environment, but it's unsafe for production environments. In production, you'll authorise only a specific IP address or range of addresses to access your instance.

NOTE: I'm not entirely certain of what the correct settings are for SSH to allow Forge to connect. For now, the only way I know is to open SSH access from "Anywhere", but I've messaged Taylor to see if there's a better configuration that both allows Forge access, but locks down your SSH access a bit.

If, later, you're creating a multiple-instance application stack, you will be able to set a Custom IP of "this security group" and allow any instances within this server to talk to each other.

10. Download key pair

You might be familiar with SSH key authentication. AWS uses the .pem format, which is yet another way for you to download pieces of a security certificate for authenticating with other machines, etc. If you're not familiar, this file you're downloading will allow you to authenticate yourself to AWS without needing to type a password every time.

This is another great chance to have a specific key per project; but it's entirely up to you. You could also choose to have one key for all of your Forge accounts, one key for the entirety of your Amazon account, or whatever else. You'll see in the screenshots I created one for laravel-forge, but again, I would likely do this project-specific in the future.

AWS Key Pairs

Download the file, and place it in a location you'll remember. I created a pem directory in ~/.ssh and placed it there (~/.ssh/pem/laravel-forge.pem).

11. Review and Launch Instance

AWS Review Instance

Finally, it's time! Review everything you have set, and once you're satisfied, Launch the instance.

AWS Site Launched

Note: you can optionally click "Creating Billing Alerts" to set up notices for when you get billed over a certain amount.

Wait a bit for it to get up and running, and then go back to View Instances and check the instance. Now, down at the bottom of the screen, you'll be able to view all the important information about this instance.

AWS View Instance.

2.Installing the Basics.

Amazon EC2 makes it very easy to scale the servers that our web applications live on. This video tutorial will show you how to load up an Amazon EC2 ubuntu server and setup Apache, PHP, & MySQL on the server. I will then go through the basics of setting up the environment for a basic Laravel 5.1 installation.

https://www.youtube.com/watch?v=8ARpTKWc6lQ

Here is the cheat sheet of commands that I was using throughout the video:

————- Installing Apache ————-

$ sudo apt-get install apache2

————- Installing Latest PHP ————-

$ sudo add-apt-repository ppa:ondrej/php5

$ sudo apt-get update

$ sudo apt-get install php5 libapache2-mod-php5

————- Installing PHP Mcrypt ext. ————-

$ sudo apt-get install php5-mcrypt

————- Installing MYSQL ————-

$sudo apt-get install mysql-server

3.Installing Git Core and Laravel Git Basic Repo

————- Installing GIT ————-

$ sudo apt-get install git-core

————- Laravel GIT Repo ————-

https://github.com/laravel/laravel.git

Now instead of pasting the contents of basic repo . You clone the contents of our website into the www/[new folder you created].

4.Installing Composer

————- Installing Composer ————-

curl -sS https://getcomposer.org/installer | php

Here are a few additional resources:

This is the command I used at the beginning of the video to run apt-get update:

$sudo apt-get update --fix-missing

Here is the URL to the documentation on how to SSH into your Amazon EC2 instance:

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html

5.Changing the .htaccess file

Hopefully Your server IP will now have the basic apache webpage and when you go to IP/[new folder] , you'll see laravel install as You have arrived if you have installed basic REPO .

If you have installed our website backup . You need to create another folder named .env

APP_ENV=local

APP_DEBUG=true

APP_KEY=tPbiHYlXPrlLlKoCKwSSlBHsrapRB1ft

DB_HOST=54.68.214.69 //your content or dbconnection . Usually LOCALHOST.

DB_DATABASE=CMRL //DB you have created

DB_DATABASE2=eqmain1

DB_USERNAME=Mohan //username and password

DB_PASSWORD=MohanCEG15

CACHE_DRIVER=file

SESSION_DRIVER=file

QUEUE_DRIVER=sync

MAIL_DRIVER=smtp

MAIL_HOST=mailtrap.io

MAIL_PORT=2525

MAIL_USERNAME=null

MAIL_PASSWORD=null

MAIL_ENCRYPTION=null

save it as **.env** file in the main www/[new folder] itself. then change mod of the whole folder acc to the server to -777, with chmod permissions.

Connect to db with sequel pro or any other db connecter in your laptop . Since your website will be in IP/laravel folder , we change the htaccess file to point it to the IP.

GO to etc/apache2/sites-avaiable. select 000-default.conf and change the document root from IP to IP/[newfolder]/public.

Now restart Apache. And it should work.