

## Documentation

Laravel has wonderful documentation covering every aspect of the framework. Whether you are a newcomer or have prior experience with Laravel, we recommend reading our documentation from beginning to end. →



## Laracasts

Laracasts offers thousands of video tutorials on Laravel, PHP, and JavaScript development. Check them out, see for yourself, and massively level up your development skills in the process. →



## Laravel News

Laravel News is a community driven portal and newsletter aggregating all of the latest and most important news in the Laravel ecosystem, including new package releases and tutorials. →



## Vibrant Ecosystem

Laravel's robust library of first-party tools and libraries, such as [Forge](#), [Vapor](#), [Nova](#), [Envoyer](#), and [Herd](#) help you take your projects to the next level. Pair them with powerful open source libraries like [Cashier](#), [Dusk](#), [Echo](#), [Horizon](#), [Sanctum](#), [Telescope](#), and more.

**Beyond Horizons** - Build your dream app with **aMiSTACK**.

## Congratulations!

And welcome to your Premium **Laravel** stack deployment by an **aMiSTACX G6F**.

As this stack was designed to be as automated as possible, with the least number of steps required to get you up and running quickly, please follow the directions closely to ensure success.

It is best advised to get the product you purchased running per this documentation first! Then you have the option to customize your solution to your requirements.

These instructions for our stack assume the following:

- You have a **Basic** understanding of the AWS console
- You have an **Intermediate/Advanced** skill level and/or experience with a Linux stack and **Laravel**.
- You have a remote access SSH client, such as putty, and you understand how to create a ppk file from an AWS PEM file. These credentials will allow you to connect to your new aMiSTACX instance in your AWS availability zone.

WinSCP sudo: <https://amistacx.io/winscp-sudo-access-for-ubuntu-amistacx>

Putty to AWS: <https://amistacx.io/how-to-use-putty-to-connect-to-aws>

How to generate a PPK file: <https://amistacx.io/how-to-generate-a-ppk-file-for-ssh-and-sftp>

Create AWS Key: <https://amistacx.io/how-to-create-an-aws-ssh-key-pair>

More Info on Putty/AWS: <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html>

AWS Web Connect: <https://amistacx.io/aws-console-ssh-web-connect>

## **G6F-[Flexibility!!]**

Introducing our new G6F stack that has both Apache and NGINX ready to go. In this way, advanced users can implement NGINX should they feel the need. Apache is the default as it is what we recommend for stability, easy maintenance, and very good performance.

### **Apache or NGINX**

Apache is enabled by default. For advanced admins you can switch to NGINX.

#### **Stop/Disable**

```
sudo systemctl stop apache2  
sudo systemctl disable apache2
```

#### **Start/Enable**

```
sudo systemctl enable apache2  
sudo systemctl start apache2
```

#### **Stop/Disable**

```
sudo systemctl stop nginx  
sudo systemctl disable nginx
```

#### **Start/Enable**

```
sudo systemctl enable nginx  
sudo systemctl start nginx
```

## What's New in v1.0 G6F

- Laravel 13.x

# I. Ubuntu 22.04 LTS Essentials

## Core Software Versions

- Ubuntu 22.04
- Apache 2.4.52
- NGINX 1.26.1
- MySQL 8.0.39
- PHP 8.3.11 [Default]
- Laravel 13.7
- Redis 7.2.2
- phpMyAdmin 5.1.3

## II. FPM/PHP Memory Allocation & Settings

**Note:** Our stack is optimized for EC2 t3-small. You will need to adjust these settings for larger instances to achieve maximum performance.

**Note:** FPM is running under [www-data:www-data](#) [This means that should you deploy a web application under “/var/www/”, then it is best to utilize the www-data user/group; otherwise, you need to update the FPM pool.]

**/etc/php/8.x/fpm/pool.d/www.conf**

FPM Pool is set to **ondemand**

FPM Pool Settings for Server and Children default

```
pm.max_children = 55
pm.start_servers = 10
pm.min_spare_servers = 5
pm.max_spare_servers = 15
pm.max_requests = 500
```

**Note:** Should you run into memory issues, these settings may need to be adjusted. Should you be running a medium or large+ EC2, these settings should reflect the additional memory available.

## PHP 8.x settings

/etc/php/8.x/fpm

```
memory_limit = 2G
upload_max_filesize = 150M
post_max_size = 151M
max_execution_time = 300
```

## MYSQL [Non-default settings]

/etc/mysql/mysql.conf.d/mysqld.cnf

```
key_buffer_size      = 64M
max_allowed_packet   = 64M
thread_stack         = 193K
wait_timeout         = 60
```

**/etc/systemd/system/mysql.service.d/override.conf** #Open File Increase

```
[Service]
LimitNOFILE=65535
```

### III. AWS Security Group Confirmation

When first creating your EC2 stack, make sure your AWS security group [inbound] allows the following protocols and ports: SSH 22, HTTP\* 80, HTTPS 443 incoming, TCP 8080 [docs].

**Note:** It is recommended that you verify everything is working before changing the SSH to only allow specific connections.

Security Group: sg-bb

Description	<b>Inbound</b>	Outbound	Tags
-------------	----------------	----------	------

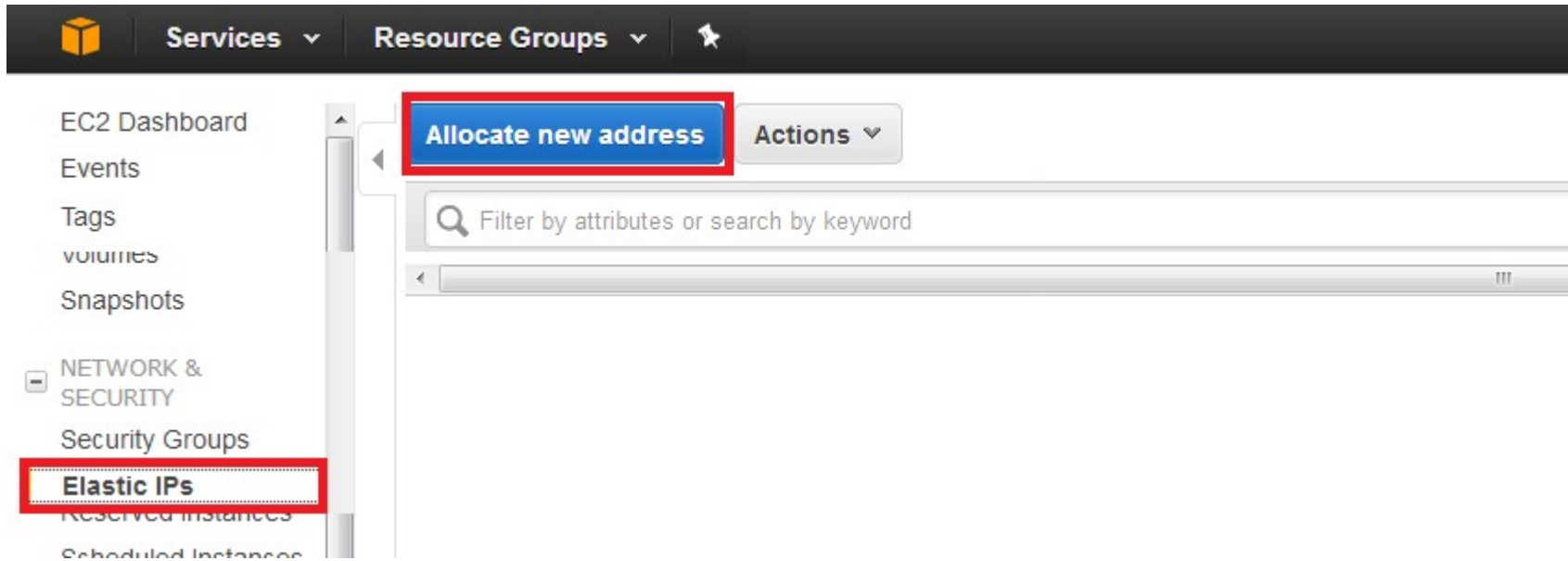
Edit
------

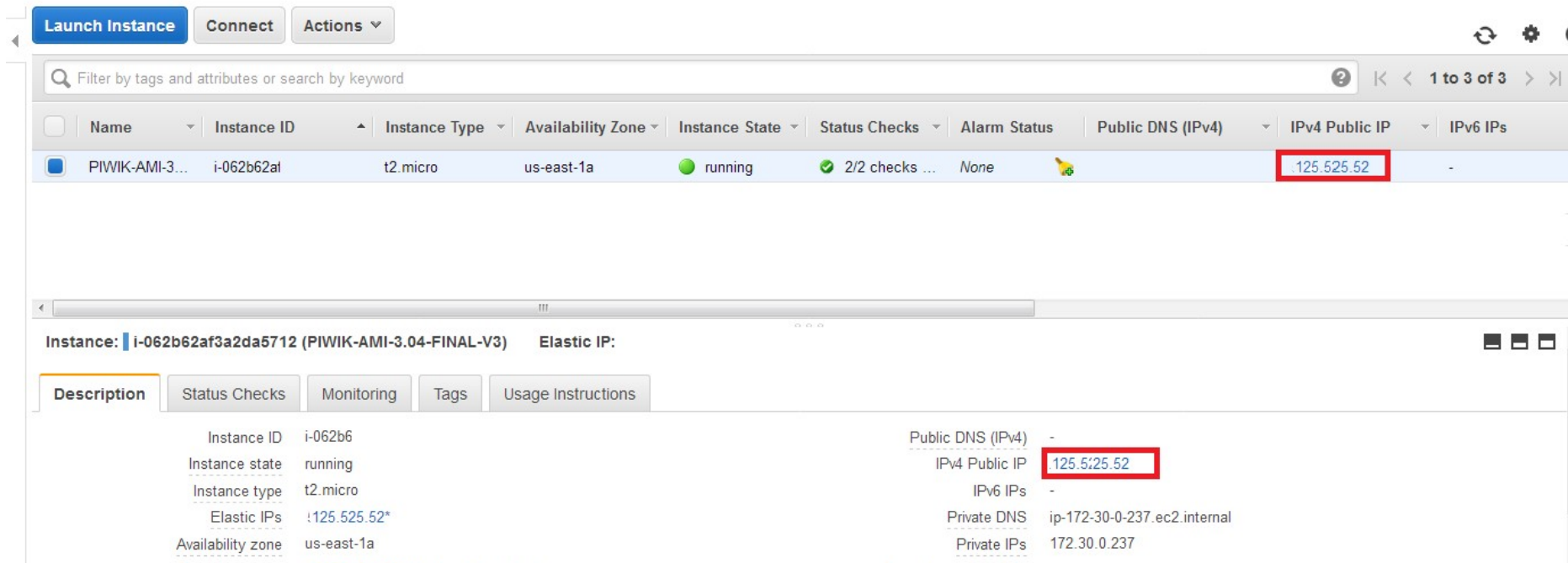
Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
HTTP	TCP	80	0.0.0.0/0
HTTP	TCP	80	::/0
SSH	TCP	22	0.0.0.0/0
HTTPS	TCP	443	0.0.0.0/0
HTTPS	TCP	443	::/0

## IV. AWS Elastic IP Address [Allocation]

It is strongly recommended that you create an AWS elastic IP address associated to this new EC2 build instance. This will allow you to start and stop without having to update public IP address connection information.



## V. AWS Public IP Address [Setting]



The screenshot shows the AWS Management Console interface for an EC2 instance. At the top, there are buttons for 'Launch Instance', 'Connect', and 'Actions'. Below this is a search bar and a table of instances. The table has columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS (IPv4), IPv4 Public IP, and IPv6 IPs. One instance is listed with the name 'PIWIK-AMI-3...', Instance ID 'i-062b62af3a2da5712', Instance Type 't2.micro', Availability Zone 'us-east-1a', Instance State 'running', Status Checks '2/2 checks ...', Alarm Status 'None', Public DNS (IPv4) '-', IPv4 Public IP '125.525.52', and IPv6 IPs '-'. The 'IPv4 Public IP' value is highlighted with a red box. Below the table, there is a detailed view for the selected instance, showing tabs for Description, Status Checks, Monitoring, Tags, and Usage Instructions. The 'Description' tab is active, showing instance details: Instance ID (i-062b62af3a2da5712), Instance state (running), Instance type (t2.micro), Elastic IPs (125.525.52\*), and Availability zone (us-east-1a). On the right side of the detailed view, there are fields for Public DNS (IPv4) (-), IPv4 Public IP (125.525.52), IPv6 IPs (-), Private DNS (ip-172-30-0-237.ec2.internal), and Private IPs (172.30.0.237). The IPv4 Public IP value is also highlighted with a red box.

After your image is built, first confirm you can access SSH, HTTP, and HTTPS.

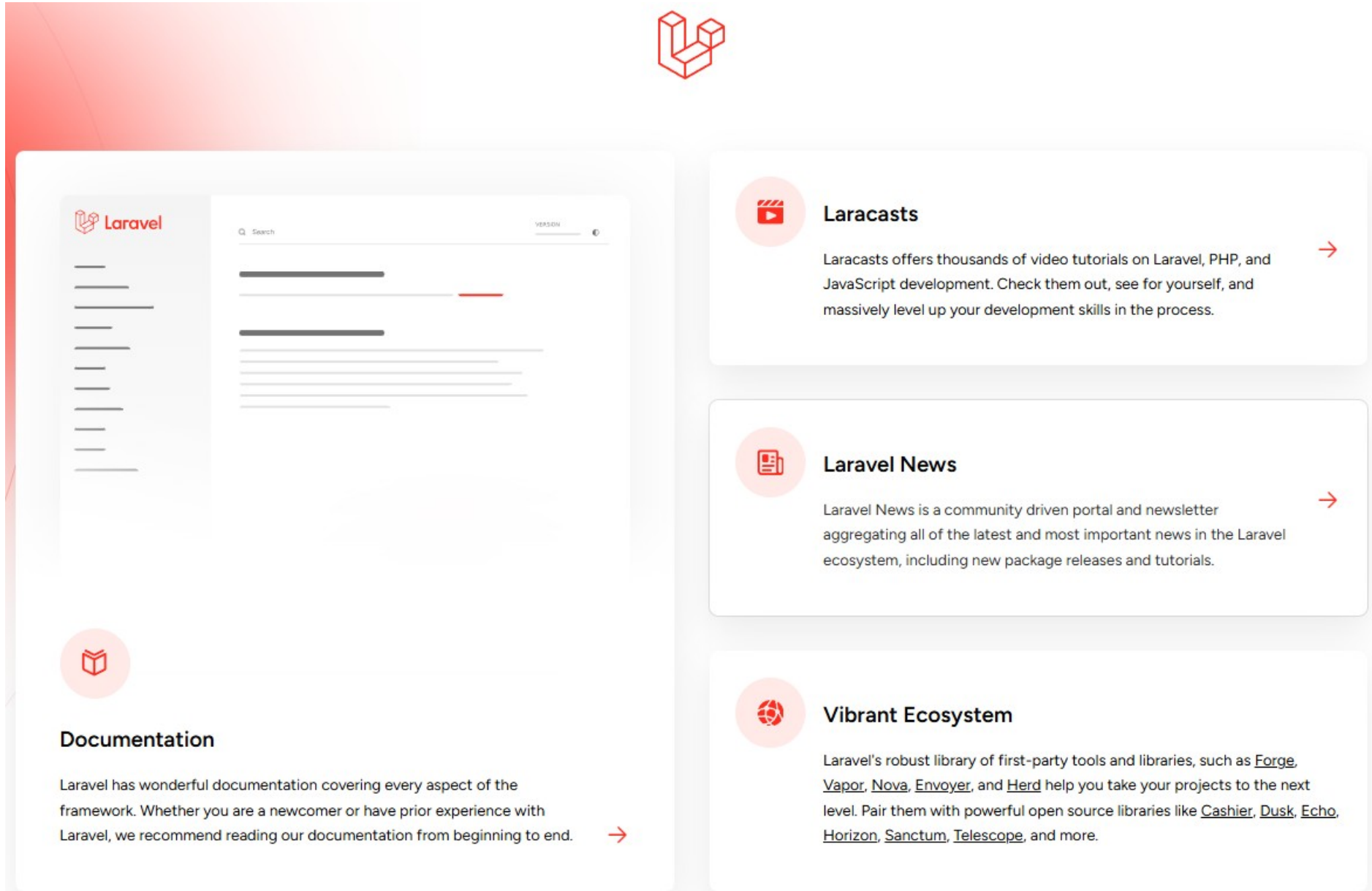
Your IP address is the elastic public IP address. You use this for DNS and for SSH.

To check HTTP: [http://<AWS\\_Public\\_IP\\_Address>/](http://<AWS_Public_IP_Address>/)

To check HTTPS: [https://<AWS\\_Public\\_IP\\_Address>/](https://<AWS_Public_IP_Address>/)

**Note:** You will need to add an exception for HTTPS as you are using a self-signed cert.

## Your HTTP or HTTPS test will show the Laravel Splash Screen - Success!

A screenshot of the Laravel website splash screen. The page has a red header with the Laravel logo. Below the header, there are three main sections: 'Documentation', 'Laracasts', and 'Laravel News'. Each section has a red circular icon and a right-pointing arrow. The 'Documentation' section includes a brief description of the framework's documentation. The 'Laracasts' section describes video tutorials. The 'Laravel News' section describes a community-driven portal and newsletter. The 'Vibrant Ecosystem' section lists various tools and libraries available for Laravel.

**Documentation**

Laravel has wonderful documentation covering every aspect of the framework. Whether you are a newcomer or have prior experience with Laravel, we recommend reading our documentation from beginning to end. →

**Laracasts**

Laracasts offers thousands of video tutorials on Laravel, PHP, and JavaScript development. Check them out, see for yourself, and massively level up your development skills in the process. →

**Laravel News**

Laravel News is a community driven portal and newsletter aggregating all of the latest and most important news in the Laravel ecosystem, including new package releases and tutorials. →

**Vibrant Ecosystem**

Laravel's robust library of first-party tools and libraries, such as [Forge](#), [Vapor](#), [Nova](#), [Envoyer](#), and [Herd](#) help you take your projects to the next level. Pair them with powerful open source libraries like [Cashier](#), [Dusk](#), [Echo](#), [Horizon](#), [Sanctum](#), [Telescope](#), and more.

## VI. DNS Cloudflare

### Cloudflare [\*Recommended\* Easy to configure]

Our instructions use DNS/CDN provider Cloudflare for examples, and is recommended for users with basic to intermediate Administration/Networking skills.

CF offers a great easy to use DNS service, that is very user friendly, is **Free** to use for basic features. It's a great starting point to get up and running quickly!

<https://www.cloudflare.com/plans/>

**Note:** The Cloudflare Free plan has a restriction of **100MB** file uploads through their CDN. You can use Cloudflare for DNS only, but if you require file uploads on your site from your customers that exceed 100MB, then you will have to upgrade to a paid plan.

**Tip:** A51 can make use of the Cloudflare API for simple CDN management: Purge cache and ON/OFF. A helpful tool during development.

## VII. Recommended Stack Configurations [Optional - For advanced Linux Users]

**Note:** Should you want to use a DNS friendly name and real SSL cert, follow directions in this section; otherwise, you may proceed with the next section.

### Apache Friendly DNS Name w/ Domain or Subdomain

In conjunction with external DNS, if want you to use a friendly name, you will need to access the server via SSH and use the ubuntu user to sudo to update the following:

#### 1A. Subdomain: [Example. [www.example.com](http://www.example.com)]

**sudo nano /etc/apache2/sites-available/laravel.conf**

Un-comment line “remove #” and update to ServerAlias [subdomain.example.com](http://subdomain.example.com) [where example.com = your domain name]

**sudo nano /etc/apache2/sites-available/laravel-ssl.conf**

Un-comment line “remove #” and update to ServerAlias [subdomain.example.com](http://subdomain.example.com) [where example.com = your domain name]

**Save files!** And from from CLI: **sudo service apache2 restart**

#### 1B. Point external A record DNS to your new subdomain > [subdomain.example.com](http://subdomain.example.com)

## 2A. Domain: [Example. example.com]

**sudo nano /etc/apache2/sites-available/laravel.conf**

Un-comment line “remove #” and update to ServerName **example.com** [where example.com = your domain name]

**sudo nano /etc/apache2/sites-available/laravel-ssl.conf**

Un-comment line “remove #” and update to ServerName **example.com** [where example.com = your domain name]

**Save files!** And from from CLI: **sudo service apache2 restart**

## 2B. Point external A record DNS to your new domain > **example.com**

## NGINX Friendly DNS Name w/ Domain or Subdomain

### 1A. Subdomain: [Example. subdomain.example.com]

`sudo nano /etc/nginx/sites-available/laravel`

```
9 ### SSL configuration
10 ### http1 and http2
11 server {
12     #listen 443 ssl;
13     #listen [::]:443 ssl;
14
15     listen 443 ssl http2;
16     listen [::]:443 ssl http2;
17
18     server_name www.example.com example.com;
19
20     ### Magento Document Root
21     set $MAGE_ROOT /var/www/magento;
22     include /var/www/magento/nginx.conf.sample;
23     index index.html index.php;
24
25     ### Decide if you want to use Let's Encrypt for Certificates
26     include /etc/nginx/snippets/letsencrypt.conf;
```

Update to server\_name **subdomain.example.com** [where example.com = your domain name]

e.g.

server\_name **www.example.com**;

**Note:** Put the server names to listen on in each sever block sections of HTTPS and HTTP.

**Save file!** And from from CLI: `sudo service nginx restart`

**1B. Point external DNS A record to your new subdomain > [subdomain.example.com](#)**

**2A. Domain: [[Example. example.com](#)]**

Update to server\_name **example.com** [where **example.com** = your domain name]

**server\_name [example.com](#);**

**Note:** Put the server names to listen on in each sever block sections of HTTPS and HTTP.

**Save file!** And from from CLI: **sudo service nginx restart**

**2B. Point external DNS A address to your new domain > [example.com](#)**

## VIII. TLS/SSL [HTTPS] Configuration [Optional]

There are many ways to proceed with implementing HTTPS on aMiSTACX. For the purpose of this article, we will discuss four basic options: Free Self-Signed Placeholder, Cloudflare Free Origin Certificates, Let's Encrypt Free Wildcard Certificates, and installing a paid certificate. HTTP to HTTPS redirection is also discussed.

[How to install a TLS certificate on aMiSTACX >>](#)

## IX. MySQL 8 Connection information

**Note:** MySQL is not used in this version of Laravel. SQLite is the default since Laravel 11. You will need to enable and start mysql prior to use. If you can't log into the database for whatever reason, you will need to reset the password using the debian-sys-maint user.

Login = root

Password = your EC2 Instance ID

From AWS Web Console, or obtain via CLI: `~$ ec2metadata --instance-id`

### Example from AWS console:



**IMPORTANT!** Please store this password in a safe location as you may later change EC2 instance IDs, and forget your password.

**Note:** You would also use these very same credentials to access the database through phpMyAdmin.

In this version of Laravel, the mysql database is disabled by default. You will need to enable and start if you wish to use the phpMyAdmin utility.

[https://Your\\_AWS\\_Public\\_IP\\_or\\_Hostname:8080/phpmyadmin/](https://Your_AWS_Public_IP_or_Hostname:8080/phpmyadmin/)

## X. Email Configuration

Postfix is installed but is **not** 100% configured!

It is advised should you use our stack for WordPress, Magento, or other CMS, using an SMTP plugin that makes life a lot better and a lot easier to configure.

However, postfix allows the server to send mail in default configuration, e.g., password reset email.

**Ref:** <https://amistacx.io/aws-ec2-postfix-email-configuration-tips>

**Ref.** <https://help.ubuntu.com/community/Postfix>

**Ref:** <https://aws.amazon.com/workmail/>

## **XI. Post Install Security**

### **1. Lock-down `http{s}://<yourdomain>:8080/phpmyadmin/`**

For a production environment, it is strongly suggested you implement a second level of security on the phpMyAdmin URL by using AWS Security Group IP policies to restrict access.

### **2. SSH Security Group**

Consider restricting access to the SSH port via your AWS security group. As per the below article outlines.

<https://amistacx.io/restrict-network-access-with-aws-security-groups>

### **3. Register for A51 Monitoring & Control Dashboard**

<https://a51.amistacx.io>

## **XII. Performance Tuning**

### **Redis Configuration**

The discussion and optimization of Redis is beyond the scope of this guide.

<https://redis.io/>

Redis is pre-installed but disabled on this instance upon G6F creation.

## **XIII. How to switch PHP versions**

Helper scripts in `/var/www/utility`

## XIV. What's Next?

Be sure to check out our Laravel and main site's KB for tips and assistance.

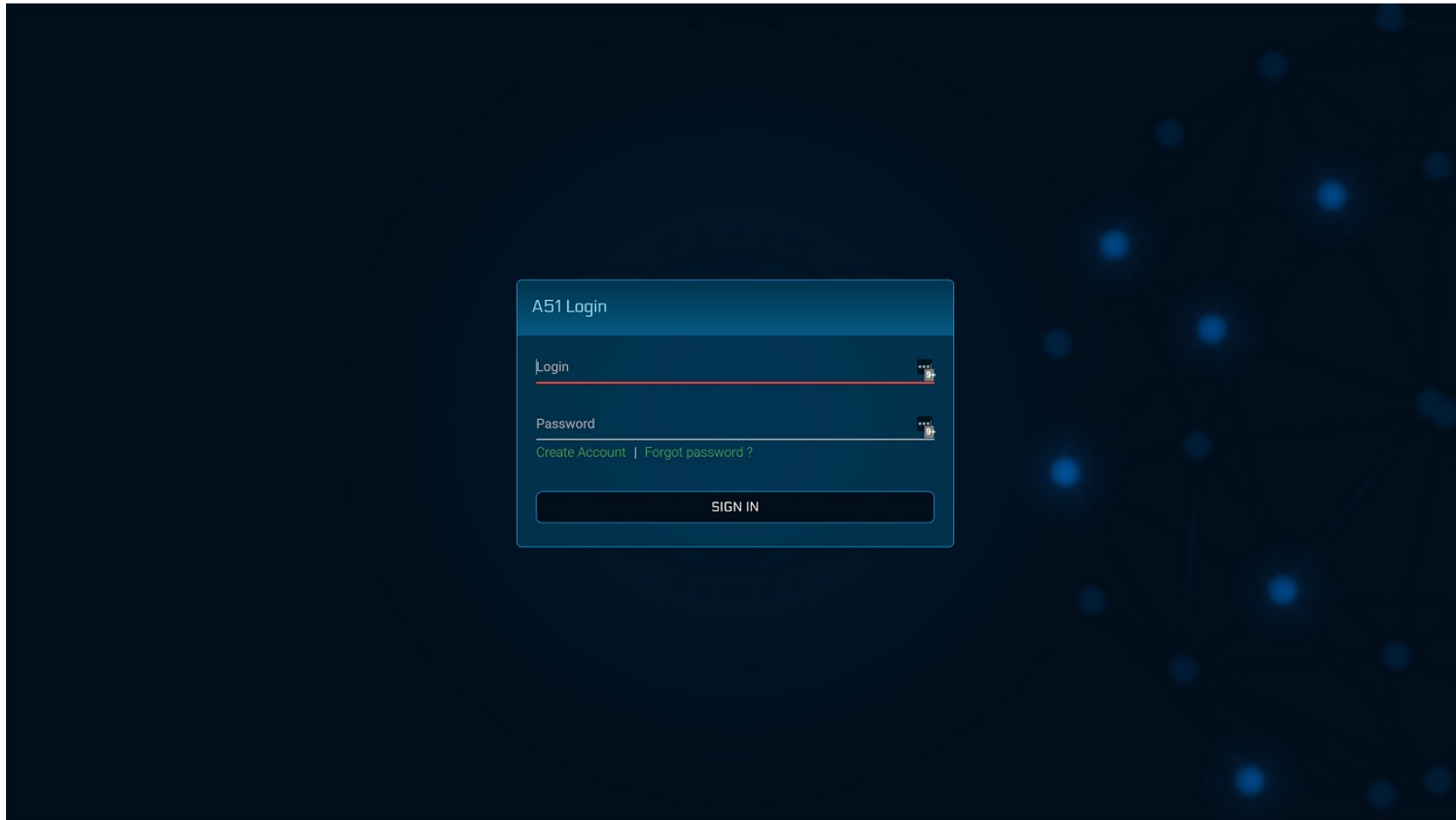
- [Register for A51](#)
- Create a FULL AMI Image/Snapshot backup
- Consider updating the Ubuntu System Files and add the latest Security patches.
- [Review our common-sense hosting tips for AWS.](#)
- [Review post deployment checklist post.](#)

**Note:** Make a full backup first!

### **Baseline Permissions:**

```
sudo chown -R www-data:www-data /var/www/laravel  
sudo chmod -R u+rwX,go+rX,go-w /var/www/laravel
```

## XV. A51 Dashboards [Registration]



A51 dashboards will allow a centralized external management of aMiSTACX resources on AWS. You must have aMiSTACX EC2 servers in order to make use of the A51 dashboard product.

Simply click “Create Account” from the login screen and follow the onscreen prompts.

More details and updates can be found at <https://amistacx.io/a51-management-console-for-aws>

A51 Guide: [https://s3.ca-central-1.amazonaws.com/amistacx.io/mp/stacx\\_a51/A51-dashboards-documentation.pdf](https://s3.ca-central-1.amazonaws.com/amistacx.io/mp/stacx_a51/A51-dashboards-documentation.pdf)

## **XIV. A51 Advanced Monitoring & Alerting**

If you deployed your stack with the AWS CloudWatch Agent, it is now available. Please review the following for usage, and we have videos on our Y/T channel. If you did not install, there is an install script in `/var/www/utility/` should you want to install it at a later time.

<https://amistacx.io/aws-ec2-and-rds-alerting-and-monitoring>

<https://amistacx.io/enable-cloudwatch-agent-for-a51>

## XVII. Support

Should you need help or have questions, please reach out to support. We will do our best to respond within 24hrs, and if you can't wait you can try our AI [MaceyBot](#). She's available 24/7/365.

Home & KB: <https://amistacx.io>

YouTube Channel: <https://www.youtube.com/@Turnkey-Ecommerce>

Thanks for selecting **aMiSTACX** as your Premium AWS EC2 stack provider. **Better - Stronger - Faster!**

