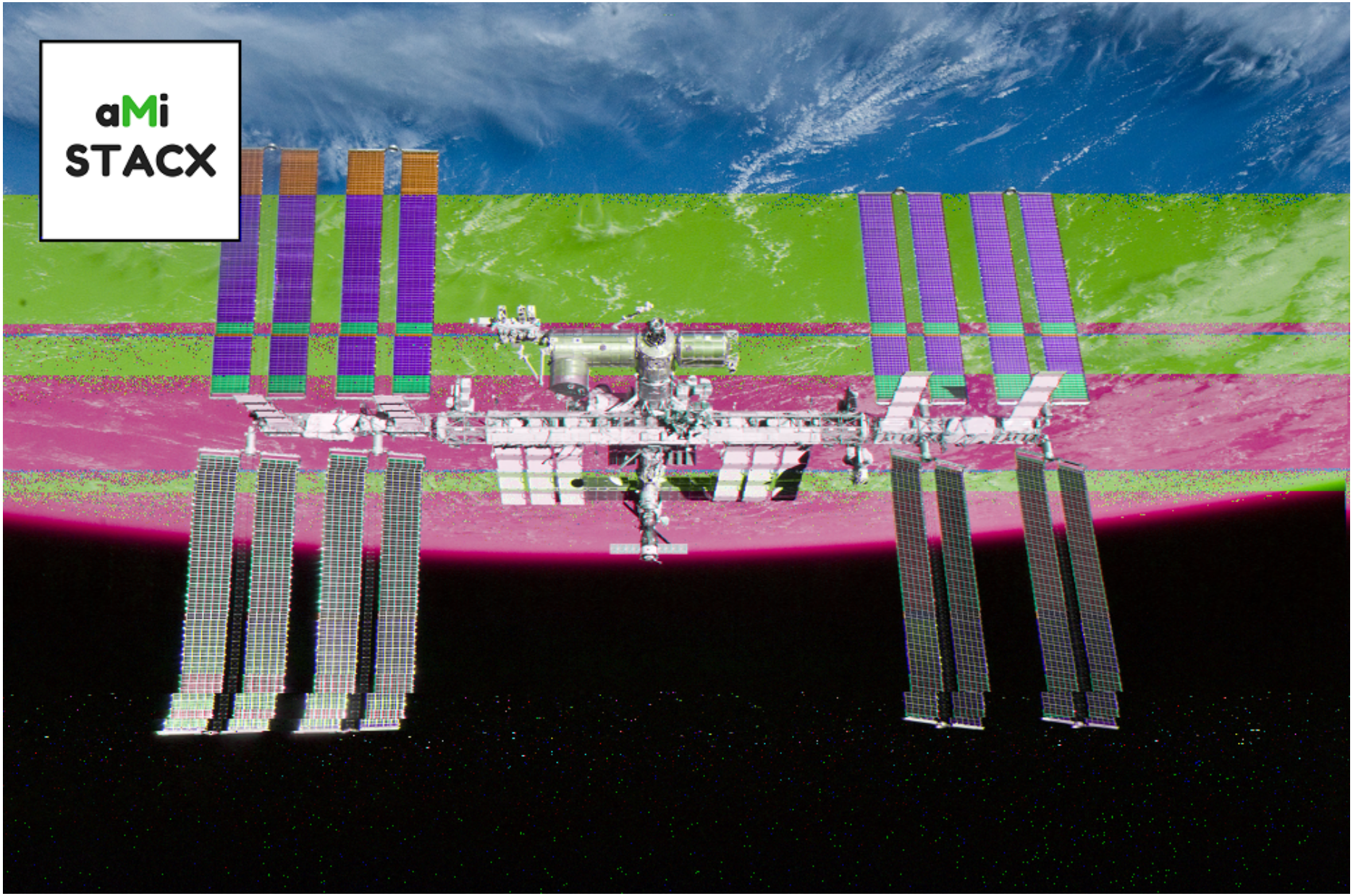


Beyond Horizons - Build your dream app with **aMi STACX.**

Congratulations!

And welcome to your Premium **Laravel 5.5.x** stack powered by **aMi STACX**.

Important! As a supported managed service aMi STACX will install and configure this AWS AMI stack for you should you not have enough experience with AWS hosting. Please contact support@amistacx.com to schedule.

or

DIY - As this stack was designed to be as automated as possible, with the least amount 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 an **Basic** understanding of the AWS console
- You have an **Intermediate/Advanced** skill level and/or experience with a Linux stack and **Laravel**.
- You have 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 aMi STACX instance in your AWS availability zone.

Putty SSH AWS: https://amistacx.com/aws-ami-stacx-connect-aws-putty.html

WinSCP AWS Sudo: https://amistacx.com/aws-ami-stacx-winscp.html

What's New in v1.4

- **Ubuntu Security Roll-up**

Note: See change log for full release update information.

https://amistacx.com/mp/stacx_change_logs/ami_stacx_laravel_5.5.0_install_change_log_v1.1.txt

I. Ubuntu 16.04.3 LTS Essentials

Core Software Versions

- **Ubuntu 16.04.3**
- **Apache 2.4.18**
- **MySQL 5.7.20**
- **PHP 7.0.22**
- **Laravel 5.5.0**

II. FPM/PHP Memory Allocation & Settings

Note: Our stack is optimized for EC2 t-micro and t-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/7.0/fpm/pool.d/www.conf

FPM Pool is set to **ondemand** [This is to help t-micro and t-small Instances]

FPM Pool Settings for Server and Children default

pm.max_children = 32
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.

/etc/apache2/conf-enabled/php7.0-fpm.conf

Idle time set to 180. For really long-running scripts, you may need to adjust this higher; however, too high won't kill a runaway script soon enough.

/etc/apache2/mods-enabled/mpm_prefork.conf

MaxRequestWorkers 256

PHP 7.x settings

/etc/php/7.0/fpm
/etc/php/7.0/apache2

memory_limit = 256M
upload_max_filesize = 50M
post_max_size = 51M
max_execution_time = 90

MYSQL [Non-default settings]

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

key_buffer_size = 32M
max_allowed_packet = 32M
thread_stack = 193K
wait_timeout = 60

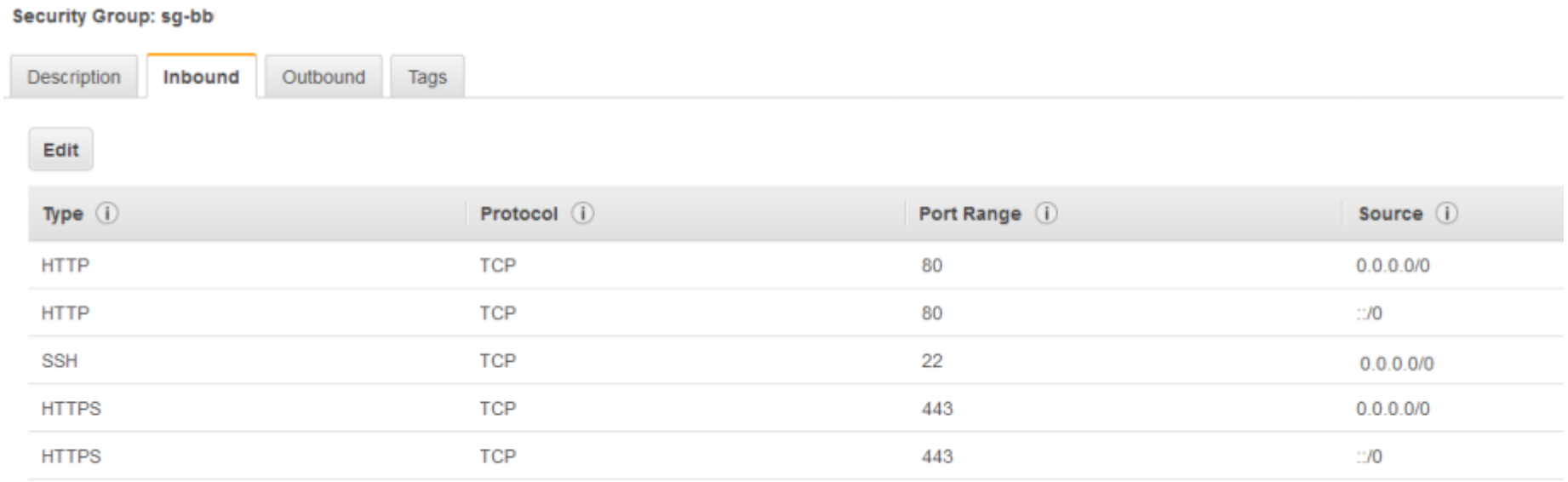
Ubuntu System

Swap File = 1MB [Not proportional to instance size. Should you deploy a stack larger than a t-small, then it suggested you adjust your swap file no more than 2x RAM.]

III. AWS Security Group

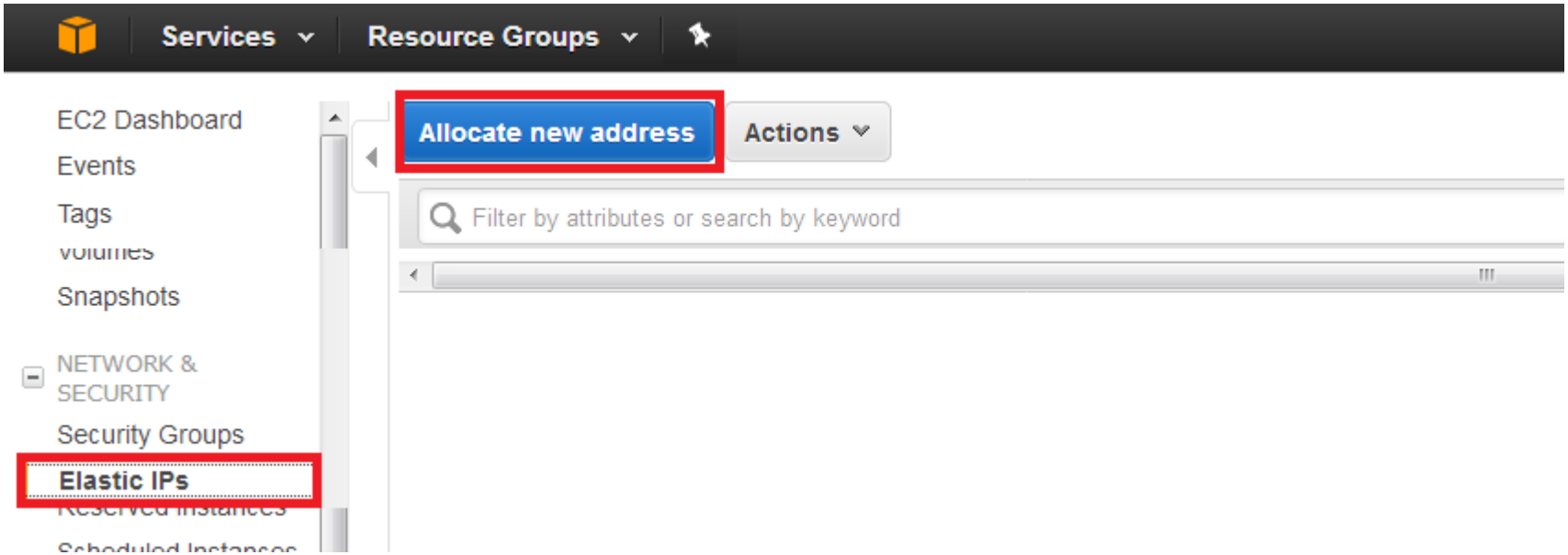
When first creating your EC2 stack, make sure your AWS security group allows the following protocols: SSH, HTTP, HTTPS incoming. The EC2 setup wizard should set these ports by default upon aMi STACX EC2 launch; however, it is good practice to confirm these ports are open during initial setup.

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

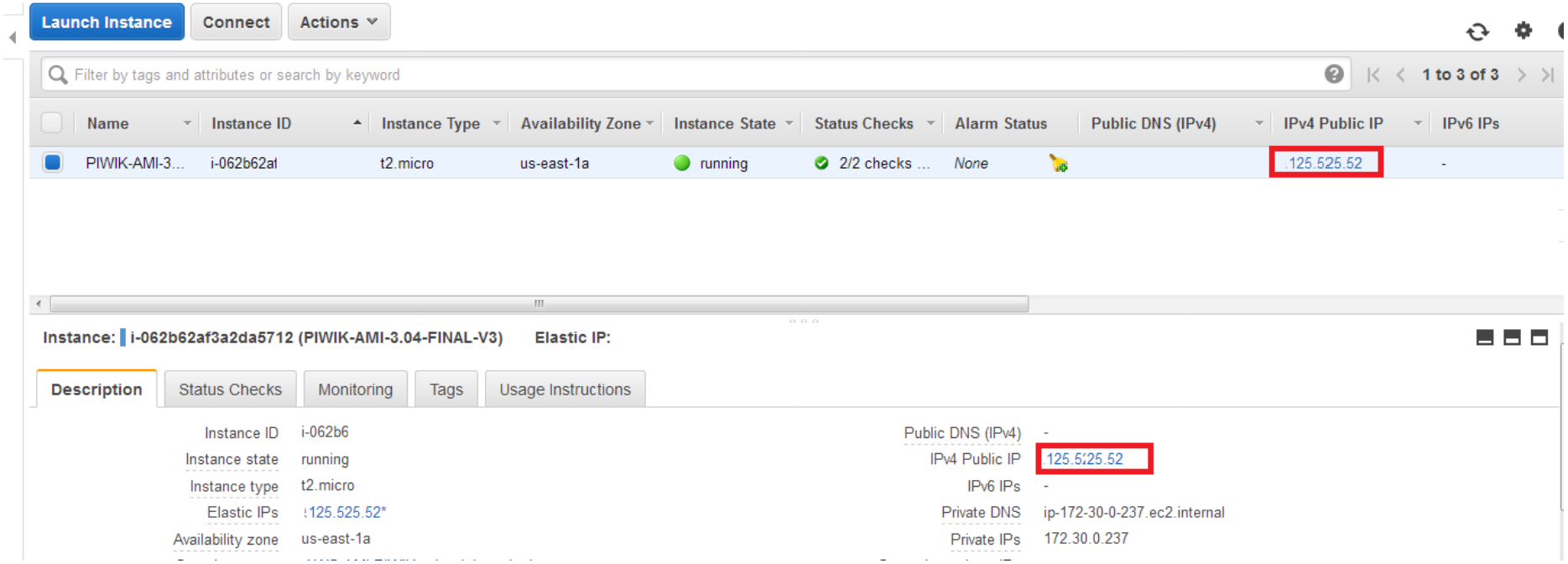


IV. AWS Elastic IP Address

If you have not done so already, it is recommended that you create an AWS elastic IP address associated to this new EC2 build instance. This will allow you to start, stop, and reboot without having to update the public IP address connection information.



V. AWS Public IP Address



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>/**

To check HTTPS: **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!

VI. DNS CloudFlare

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, and removes the complexity that Amazon's AWS Route 53 introduces.

It's a great starting point to get up and running quickly!

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

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.

Friendly DNS Name w/ Domain or Subdomain

[**Note:** It is important that you consider using a subdomain as you may want to use a free certificate(s) from Let's Encrypt and they do not offer wildcard certs - yet! This stack is configured to use Let's Encrypt. For example, www is considered a subdomain, and if you wanted to use both www and yourdomain.com you would require two certificates.]

In conjunction with external DNS, if want 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.yourdomain.com]

sudo nano /etc/apache2/sites-available/yourdomain.conf

Un-comment line "remove #" and update ServerAlias to ***subdomain***.**yourdomain.com** **[where yourdomain.com = your domain name]**

sudo nano /etc/apache2/sites-available/yourdomain-ssl.conf

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

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

1B. Point external A record DNS to your new subdomain > *subdomain*.yourdomain.com

2A. Domain: [Example. yourdomain.com]

sudo nano /etc/apache2/sites-available/yourdomain.conf

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

sudo nano /etc/apache2/sites-available/yourdomain-ssl.conf

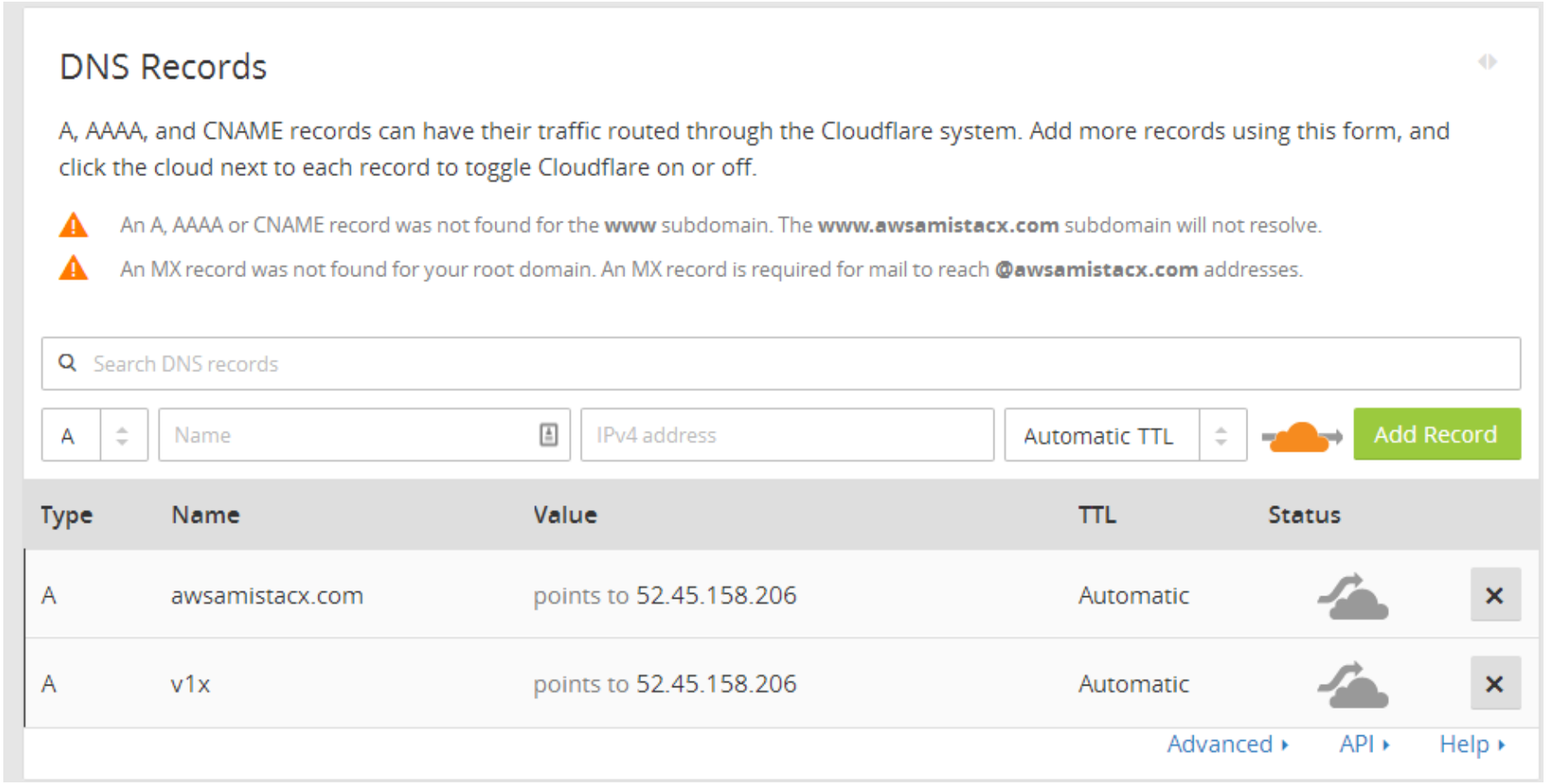
Un-comment line “remove #” and update ServerName to ***yourdomain.com*** **[where yourdomain.com = your domain name]**

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

2B. Point external DNS A record to your new subdomain > *subdomain*.yourdomain.com

e.g.

Image: CloudFlare Panel ; Note: The gray cloud **is required** at this step because we still need to get the SSL certificate; otherwise, you will get a handshake error.



VIII. SSL Configuration [Optional]

This stack comes with a dummy SSL certificate. Should you want to utilize a **Free** CloudFlare plan such as a front-run DNS and CDN provider, then you can use CloudFlare's **Full SSL** to map to the Dummy SSL without any additional SSL local configurations.

This approach is the easiest to maintain and adds the least amount of configuration and complication. See option B.

https://support.cloudflare.com/hc/en-us/articles/200170416-What-do-the-SSL-options-mean-

Let's Encrypt Option A:

Note: Before you begin - DNS must resolve! In other words, http://subdomain.yourdomain.com or http://yourdomain.com must resolve correctly to your AWS elastic public IP address. Let's Encrypt requires this otherwise the process will fail.

After confirming the DNS, to get your free certificate from let's encrypt - CLI:

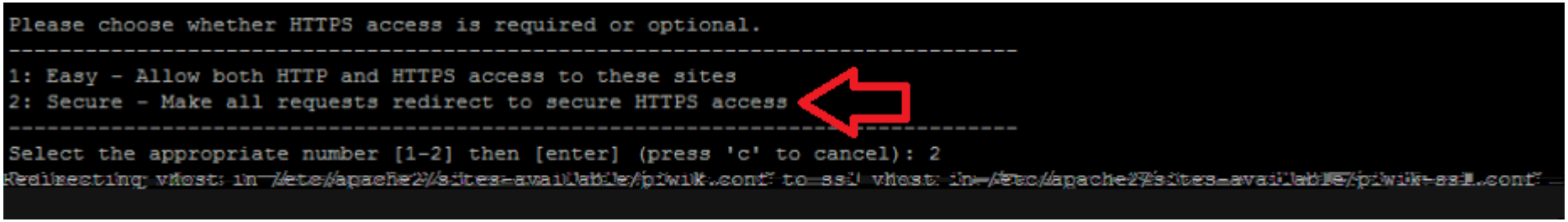
sudo certbot --apache -d *yourdomain.com*

Or

sudo certbot --apache -d *subdomain.yourdomain.com*

<continued>

Then select Option 2. Make all requests redirect to HTTPS [This places a redirect statement in yourdomain.conf]



Note: The cert expires every 90 days, you will need to set up a cron job to renew (see next page).

Note: Troubleshooting CloudFlare **TLS HandShake** Error:

Should you get a TLS handshake error, please review this article.

https://amistacx.com/ami-stacx-aws-letsencrypt-cloudflare.html

<continued>

C. Cron Setup [Required for Let's Encrypt]

To run the renewal cert check daily, we will use cron, a standard system service for running periodic jobs. We tell cron what to do by opening and editing a file called a crontab.

sudo nano /etc/crontab -e

Your text editor will open the default crontab which is a text file. Paste in the following line at the end of the file [as shown], then save and close it:

11 4 * * * root /usr/bin/certbot renew --quiet

The 11 4 * * * part of this line means "run the following command at 4:11 am, every day". You may choose any time.

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# #min hour day month weekday user command
17 *  * * * root    cd / && run-parts --report /etc/cron.hourly
25 6  * * * root  test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6  * * 7 root  test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6  1 * * root  test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
11 4  * * * root  /usr/bin/certbot renew --quiet
#

```

The renew command for Certbot will check all certificates installed on the system and update any that are set to expire in less than thirty days. --quiet tells Certbot not to output information nor wait for user input.

Note: Make sure outbound HTTP is open as the cron will poll the remote cert source via the cron job. This is normally the default from the AWS security group. Allowing ALL.

CloudFlare Option B:

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

CloudFlare is a simplistic way to have your CDN, DNS, Free SSL, and WAF all in one place. It is recommend for those that enjoy using a GUI, less complication, and ease of use.

Sign up, add your domain, have CF host your DNS, create a subdomain or domain mapping to your AWS elastic IP address. For Example, ***subdomain.yourdomain.com*** or ***yourdomain.com***

Note: This Stack is CloudFlare Aware! You do not need to make any modification to see visitors IP addresses should you use this stack for Piwik Analytics.

SSL Info: https://support.cloudflare.com/hc/en-us/articles/200170416-What-do-the-SSL-options-mean-

Step 1. After you create your DNS and point to the AWS public IP address, just make sure your CloudFlare Cert is set to FULL. This will map to your self-signed server certificates or any of your own certs that you may want to use.

Step 2. Once HTTPS is working [without getting prompted to add an exception] you'll know as your browser lock will say secured by Comodo. All you need to do is set up any HTTP to HTTPS redirect you may want to use. Many times installed applications handle this for your; otherwise, it is easiest to use CloudFlare's re-write rule. Check out our blog for examples:

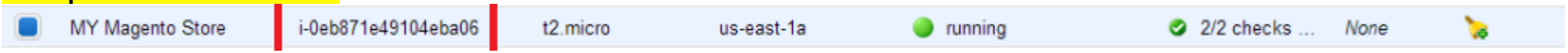
https://amistacx.com

IX. MySQL 5.7 Connection information

Login = root
Password = your AWS Instance ID

Password is 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.

http(s)://Your_AWS_Public_IP_or_Hostname/phpmyadmin

X. Email Configuration

Postfix is installed but is **not** configured! Should you have NO requirement for Postfix, please remove it.

To uninstall:

sudo apt-get remove postfix

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. ;-)

Ref.

https://help.ubuntu.com/community/Postfix

https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-postfix-on-ubuntu-16-04

XI. Post Install Security

1. Lock-down http{s}://<yourdomain>/phpmyadmin

For a production environment, it is strongly suggested you implement a second level of security on the phpMyadmin URL or move it to another port and then use AWS SG policies to restrict access.

For MFA perform the following from CLI:

A. sudo nano /etc/apache2/conf-available/phpmyadmin.conf

Uncomment: AllowOverride All

B. sudo systemctl restart apache2

C. sudo htpasswd -c /etc/phpmyadmin/.htpasswd ***username*** [You are setting username and password for the link.]

Now when you visit the /phpmyadmin you will be presented with an additional authorization box.

2. Delete the directory that contains the aMi STACX Help Documentation from:

/var/www/laravel/public/ami_stacx_{YourApplication}install/

Note: A PDF backup copy is available in your ssh user's home directory]

3. PostFix Removal [If not needed, best to remove]

Uninstall: sudo apt-get remove postfix

4. SSH Security Group Permissions

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

https://amistacx.com/aws-ami-stacx-sg-ssh.html

5. Change MySQL Root Password

The default root password = your EC2 instance ID. This is shown in the AWS console. Consider changing it to a unique value.

XII. What's Next?

Be sure to checkout our Laravel and main site's Blog for tips and assistance.

https://laravel.amistacx.com/

https://amistacx.com

XIII. Support

Should you need help or have questions, please reach out to support. We will do our best to respond within 24hrs.

Email: support@amistacx.com
Home & Blog: https://amistacx.com

Or just leave us some great feedback to let us know how we are doing. Should it not be **5-star**, please let us know how we can improve or assist you before leaving feedback.

Very Important! We may update our stacks to improve performance, address bugs, or for security upgrades. However, Amazon takes almost two weeks for new updates to be released, and we have no way to contact you!

It is strongly recommended that you checkout our website, and also check for stack updates by polling our update URL. For example, should you be on version 1.0 you can check for version 1.1 periodically. Even before Amazon releases it! It will be in this format:

https://amistacx.com/mp/stacx_change_logs/ami_stacx_laravel_5.5.**{x}** change_log_v**1**.**{0}**.txt

Just replace the version number by +1. e.g. v1.1 to see if anything has been updated. [Brackets **{}** are variable integers]

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