Elegance - Simplicity - Performance - monitor the world with **aMiSTACX.**

Congratulations!

And welcome to your Premium **Matomo v5.1.x Analytics** 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** skill level and/or experience with a Linux stack.
- 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

G-Flexibility!

Introducing our new G6F stack that has both Apache and NGINX ready to go. In this way, advanced users can implement Apache or NGINX.

Apache or NGINX

Apache is enabled by default. For advanced admins you can switch between Apache and 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 v3.7

- Ubunutu 24.04 LTS
- Matomo 5.1.x Core
- PHP 8.2 [Default]

Quick Install

Database Info

User: root
Password: {EC2 Instance ID}

Forward:

This aMiSTACX G6F for Matomo supports our module for Magento 2.x called Trak. It is also specifically designed to work with our module for Magento called Nail-the-Sale [One Page Checkout].

Read more about it here >>

Important! Do not delete the **do-not-delete.php** file found in the root if you intend to use Trak with Magento. If you do not need Trak, then you can remove the file.

I. Ubuntu 24.04 LTS Essentials

Core Software Versions

- Ubuntu 24.04
- Apache 2.4.62
- NGINX 1.26.1
- PHP 8.2.x [Default]
- MySQL 8.0.37
- Matomo 5.1.x [Upgrade after install]
- phpMyAdmin 5.1.3
- Composer 2.x

II. System and Software Configurations

Ubuntu System Settings

FPM/PHP Memory Allocation & Settings

FPM running under **www-data:www-data** and so is NGINX and Apache. This means 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.

Note: Server is configured for EC2 t3-small. You may need to adjust these settings for Maximum performance.

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

FPM Pool is set to **ondemand** [This is to help Micro and Small Instances]

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

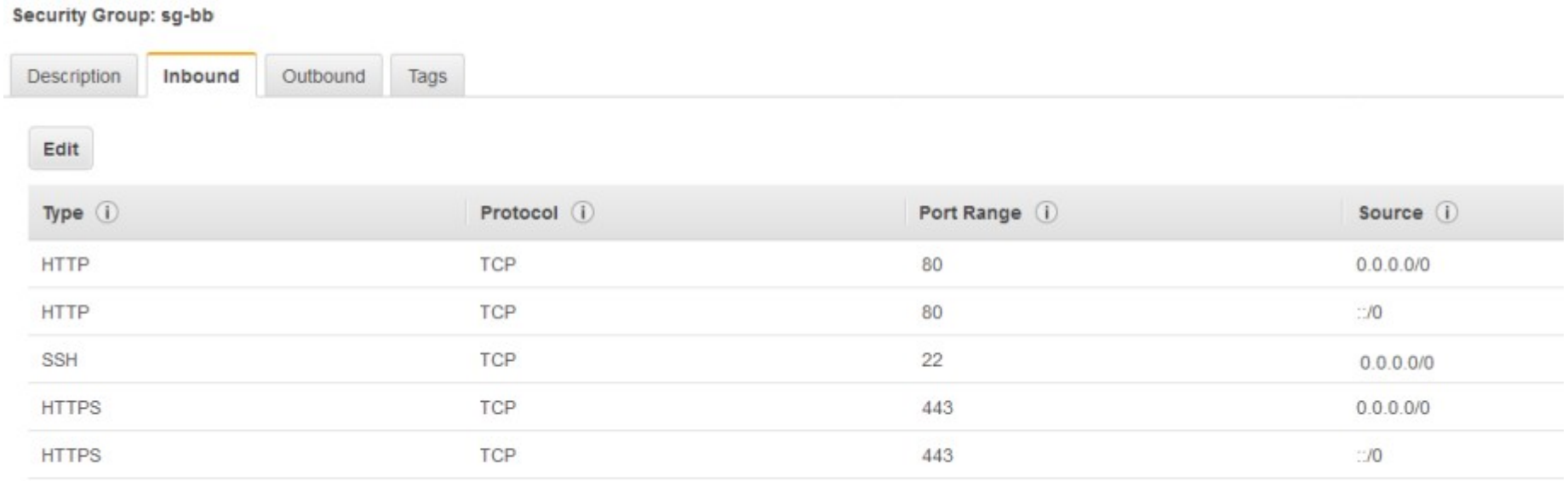
key_buffer_size = 64M
max_allowed_packet = 64M
thread_stack = 193K
wait_timeout = 300

innodb_buffer_pool_size = 1G

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.

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



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

Note: G6F server uses a secure utility port of **8080** for phpMyAdmin and documentation.

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

VI. 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]

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

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

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

Un-comment line “remove #” and update to ServerAlias ***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

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

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

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

sudo nano /etc/apache2/sites-available/matomo-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/matomo

```
### SSL configuration
### http1 and http2
server {
  #listen 443 ssl;
  #listen [::]:443 ssl;

  listen 443 ssl http2;
  listen [::]:443 ssl http2;

  server_name www.example.com example.com;

  ### Magento Document Root
        set $MAGE_ROOT /var/www/magento;
        include /var/www/magento/nginx.conf.sample;
        index index.html index.php;

  ### Decide if you want to use Let's Encrypt for Certificates
        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 >>

VIII. MySQL 8 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.

https://Your_AWS_Public_IP_or_Hostname:8080/phpmyadmin/

IX. Email Configuration

Postfix is installed but is **not** 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. ;-)

Ref.

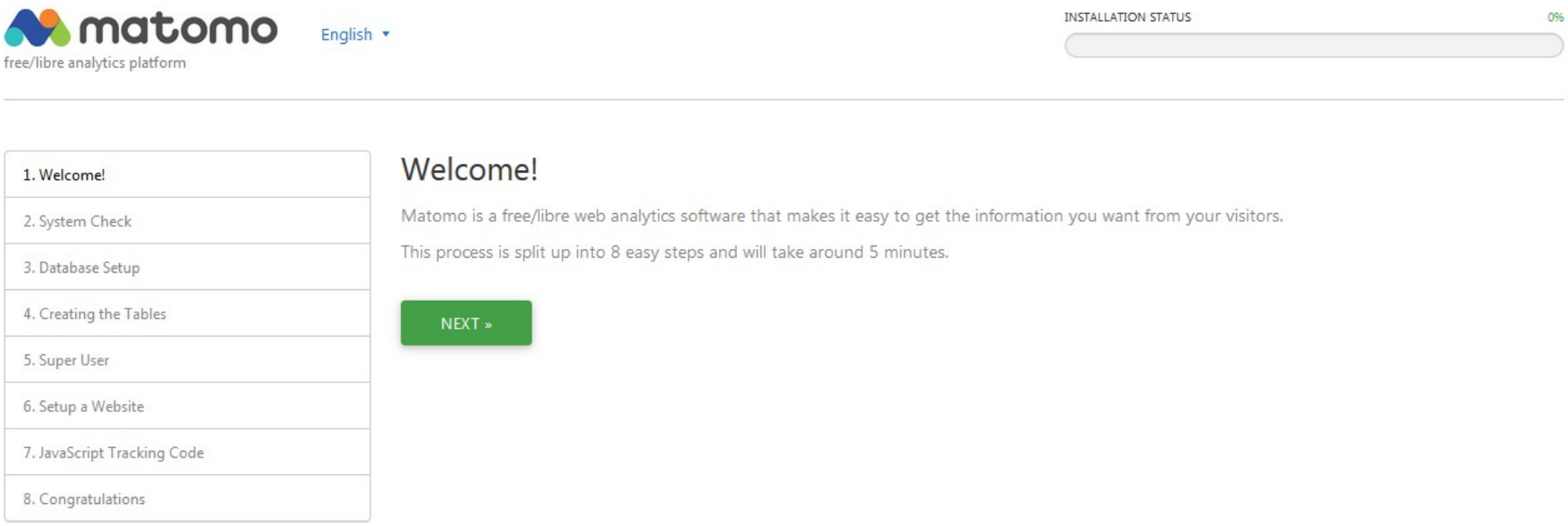
https://help.ubuntu.com/community/Postfix

https://amistacx.io/aws-ec2-postfix-email-configuration-tips

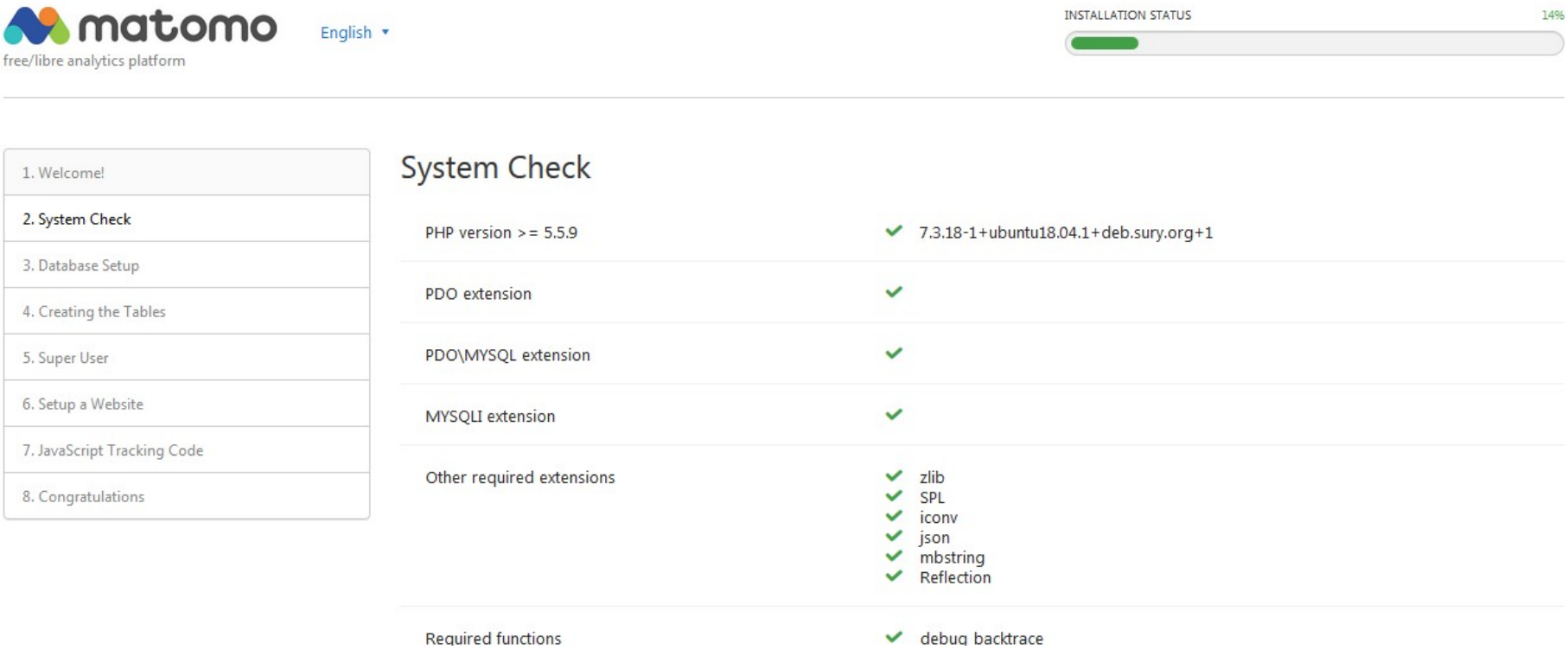
X. Matomo 5.1.x Install

https://<yourserversAWSipaddress>/

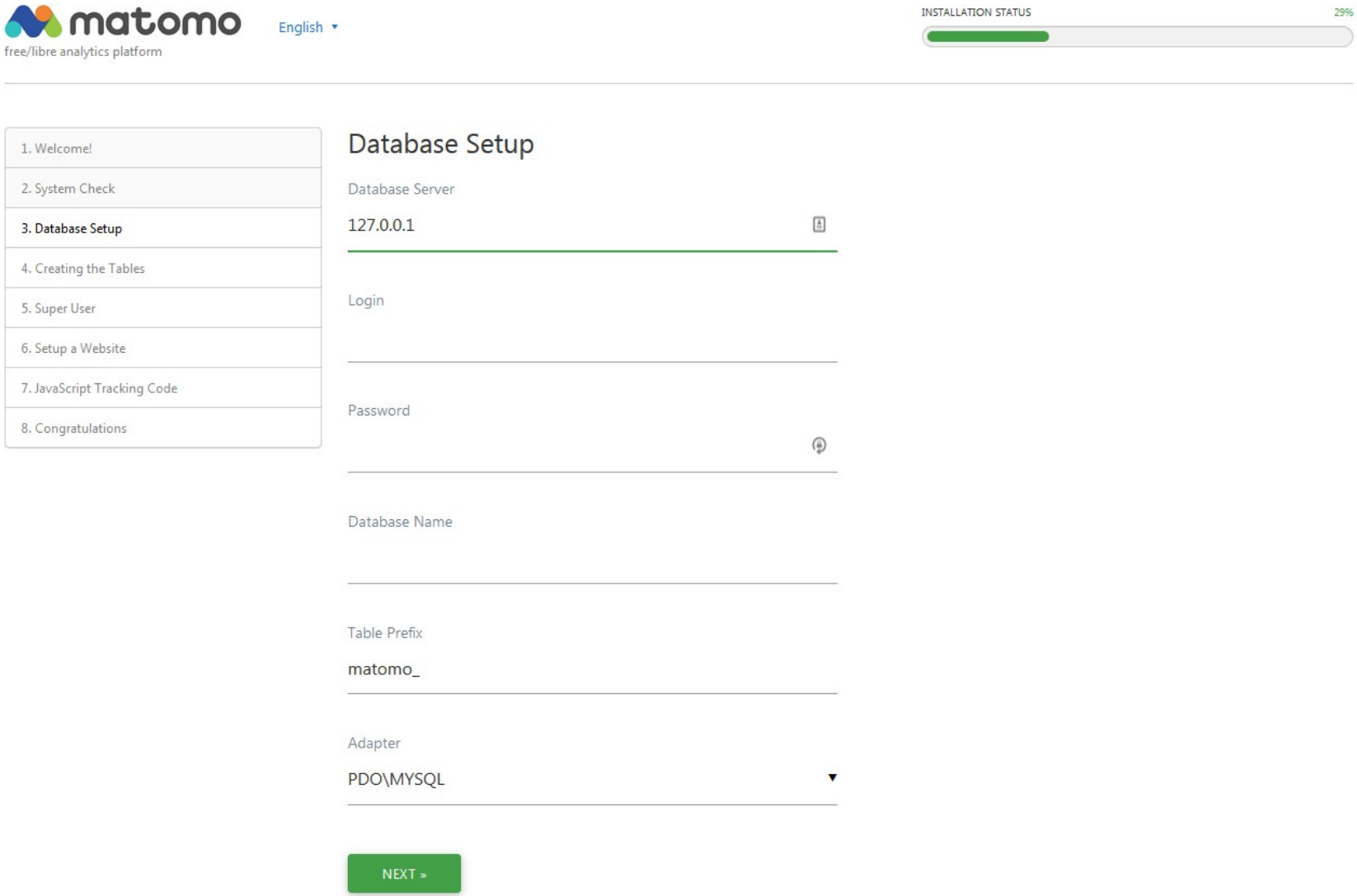
Step 1. Welcome



Step 2. System Check



Step 3. Database Config



Login = **root**
Password = **your AWS Instance ID**
Database Name = **matomo**

Password is your EC2 Instance ID. From AWS Web Console, or obtain via CLI: ~$ **ec2metadata --instance-id**

Example:

Note: Matomo 5.1.x Install Instructions [screen shots may differ than their current latest package.]:

https://matomo.org/docs/installation/#mysql-database-setup

Follow the steps from the above link, and continue to scroll down on the page for each step.

When all the steps are completed, **Matomo 5.1.x** is running!

XI. Post Install - Matomo Upgrade Option

Note: When you sign into the dashboard, should you be prompted that there is an update available, consider the options of updating at this point. However, make sure any plugins that you want to use are compatible before updating to a new version.

Note: Before you upgrade, you may want to consider making an AWS EC2 backup image of your completed stack so you do not have to go through the build process again should you encounter an issue with an upgrade.

Matomo Reports Archive Option

https://matomo.org/docs/setup-auto-archiving/

Tip! Follow our template and replace the values; otherwise, it may not work.

/etc/cron.d/matomo-archive

XII. How to switch PHP Versions

[Helper Scripts in /var/www/utility/]

XIII. 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-access-to-ssh-with-aws-security-groups

3. Register for our A51 Management Dashboard

https://a51.amistacx.io

4. Post Deployment Recommendations

https://amistacx.io/post-amistacx-deployment-checklist

XIV. What's Next?

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

https://amistacx.io/matomo

https://amistacx.io

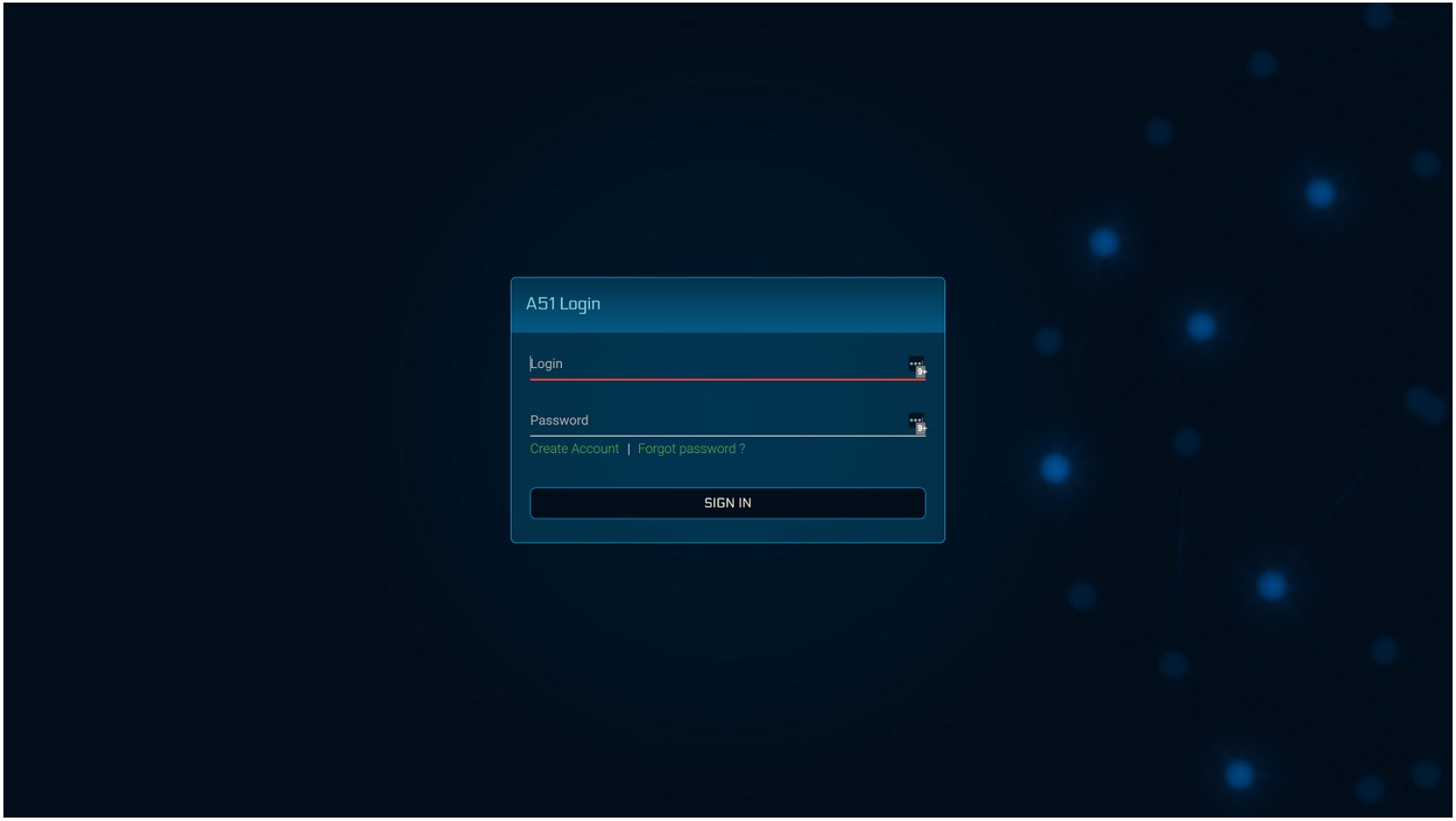
- Register for A51 Dashboards
- Create a FULL AMI Image/Snapshot backup
- Consider updating the Ubuntu System Files and add the latest Security patches.
- Review common sense AWS hosting tips.

Tip: If you need to reset your group/user permissions and folder/file permissions back to the aMiSTACX default:

sudo chmod -R u+rwX,go+rX,go-w /var/www/matomo

sudo chown -R www-data:www-data /var/www/matomo

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

XVI. A51 Advanced Monitoring

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 : https://www.youtube.com/@Turnkey-Ecommerce/videos

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