aMiSTACX G6F Welcome Page

Use this bootstrap template any way you see fit.

This site is here to show you that your **stack** is working!

aMiSTACX - **Better - Stronger - Faster!**

Elegance - Simplicity - Performance - Build your dream app with **aMiSTACX**

Congratulations!

And welcome to your Premium Developer **Ubuntu 24 LTS** stack deployment by an **aMiSTACX G6F.**

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

G6 [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

- OS Patches
- Added PHP 8.3 [Default]

I. Ubuntu 24.04 LTS Essentials

Core Software Versions

- Ubuntu 24.04 LTS
- PHP 8.3.9
- Redis 7.2.1
- NGINX 1.28
- Apache 2.4.58
- phpMyAdmin 5.2.1
- Composer 2.8.8

II. System and Software Configurations

Ubuntu System Settings

FPM/PHP Memory Allocation & Settings

FPM running under www-data:www-data [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 t3-small. You may need to adjust these settings for Maximum performance.

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

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

# Set buffer pool to 50% to 70% of available physical RAM.
innodb_buffer_pool_size = 512MB

Uncomment next line below when RAM is greater or equal to 1GB
#innodb_buffer_pool_instances = 8

Note: MySQL is configured with mysql_secure_installation with the exception of requiring strong passwords.

https://dev.mysql.com/doc/refman/8.0/en/mysql-secure-installation.html

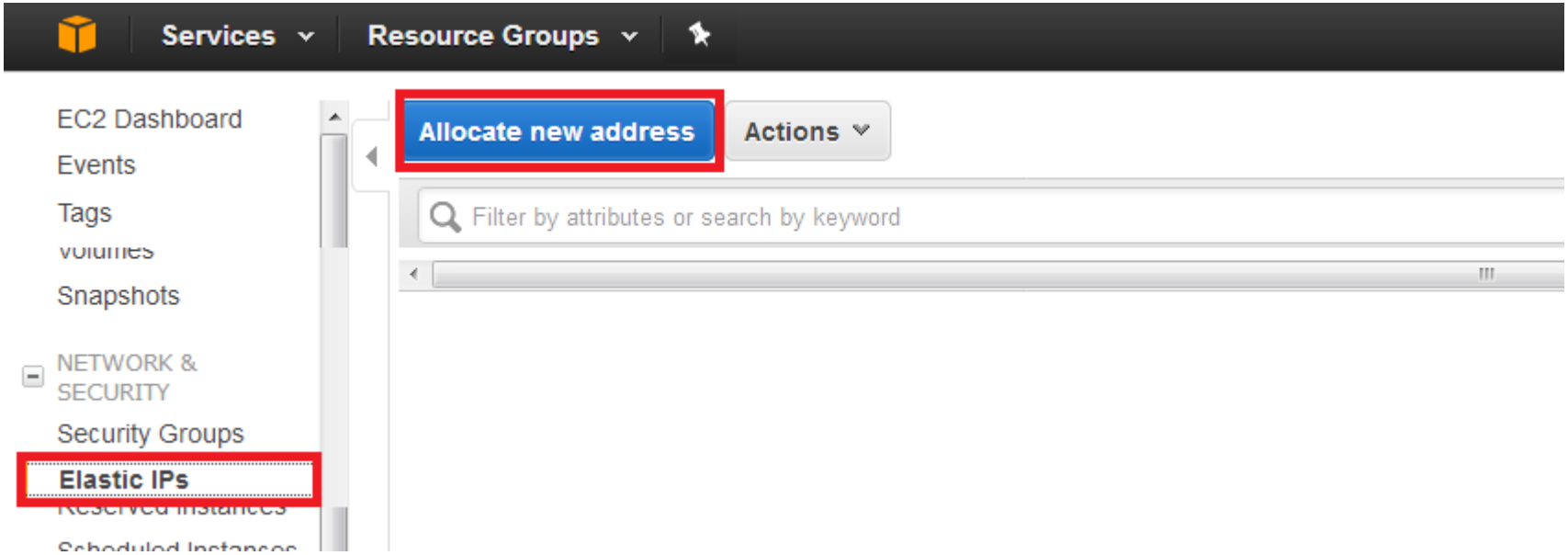
III. AWS Security Group Confirmation

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 [phpMyAdmin and Docs].

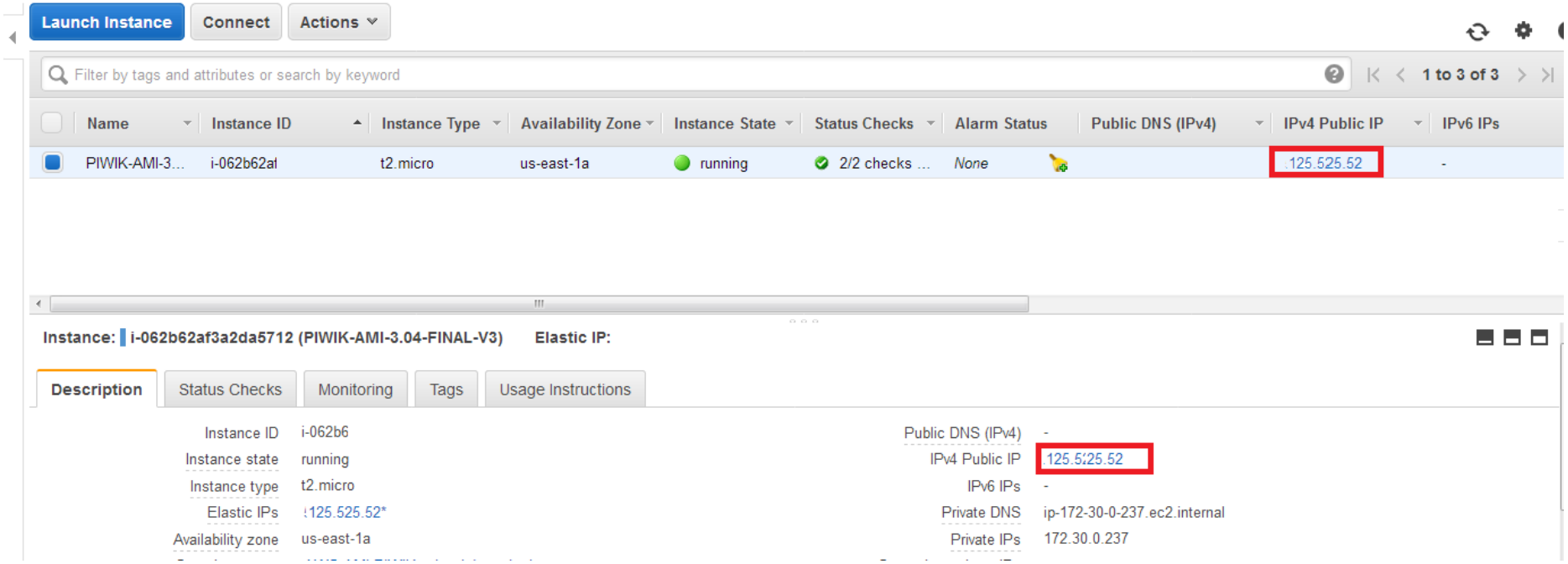
Note: It is recommended that you verify everything is working before changing the SSH to only allow specific connections.

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]



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

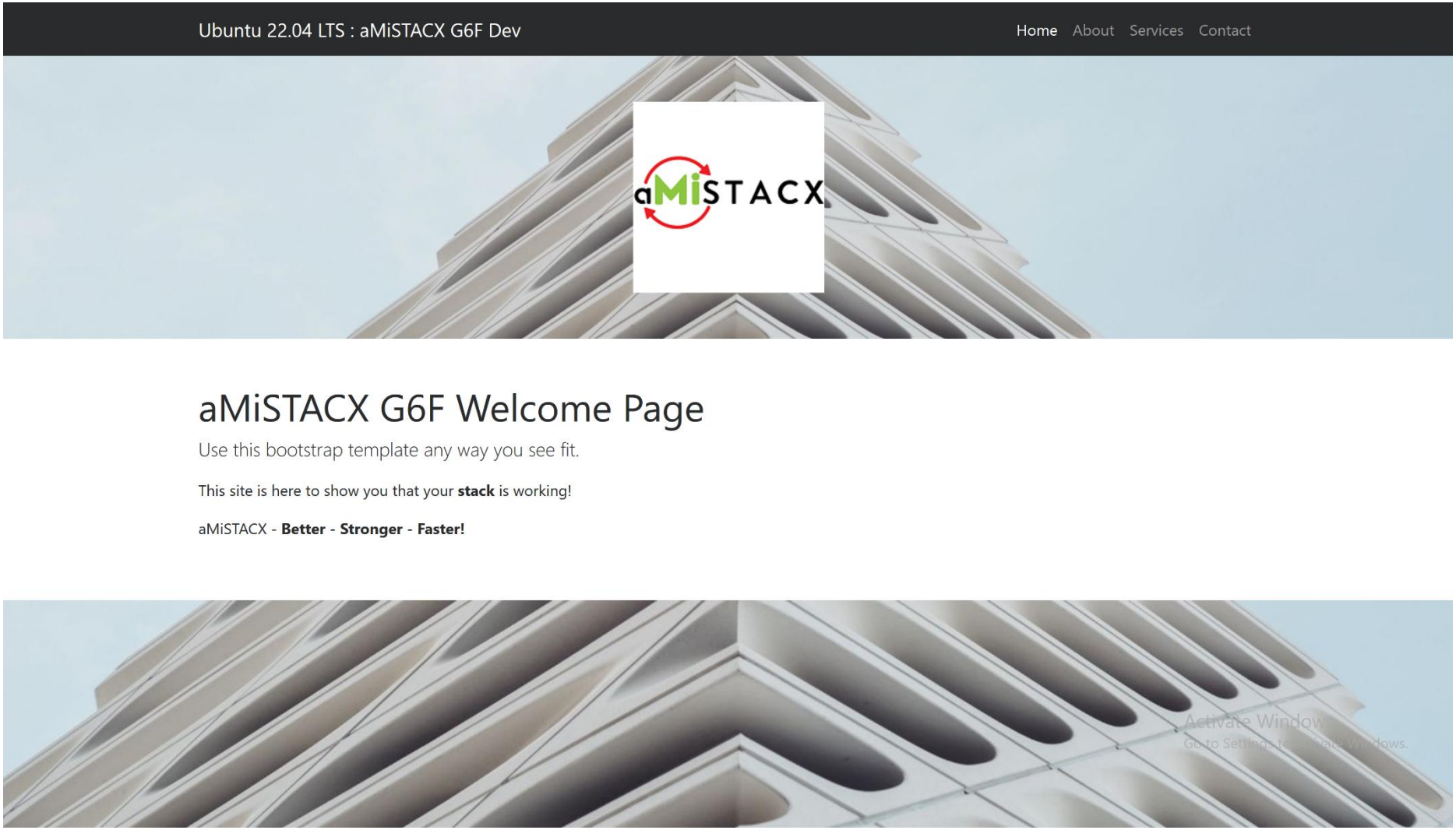
Your IP address is the elastic public IP address. You use this for DNS and for SSH.

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 server test web page should look something like this:



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]

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

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

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

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

sudo nano /etc/apache2/sites-available/domain-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/default

```
### 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.

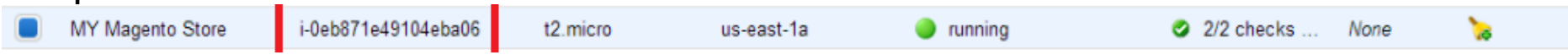
How to install a TLS certificate on aMiSTACX >>

IX. MySQL 8.0 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: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. How to switch PHP versions

Make use of the scripts in /var/www**/utility/**

XII. Misc

Part I - Redis Configuration [Caching]

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

Redis is pre-installed but NOT enabled.

To enable:

```
sudo systemctl enable redis-server.service
```

Flush:

```
sudo redis-cli flushall
```

Part II - Elasticsearch [Search]

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

ES is pre-installed but NOT enabled.

To enable:

```
sudo systemctl enable elasticsearch
```


XIII. Post Install Security Considerations

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. Post Deployment Review

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

XIV. What's Next?

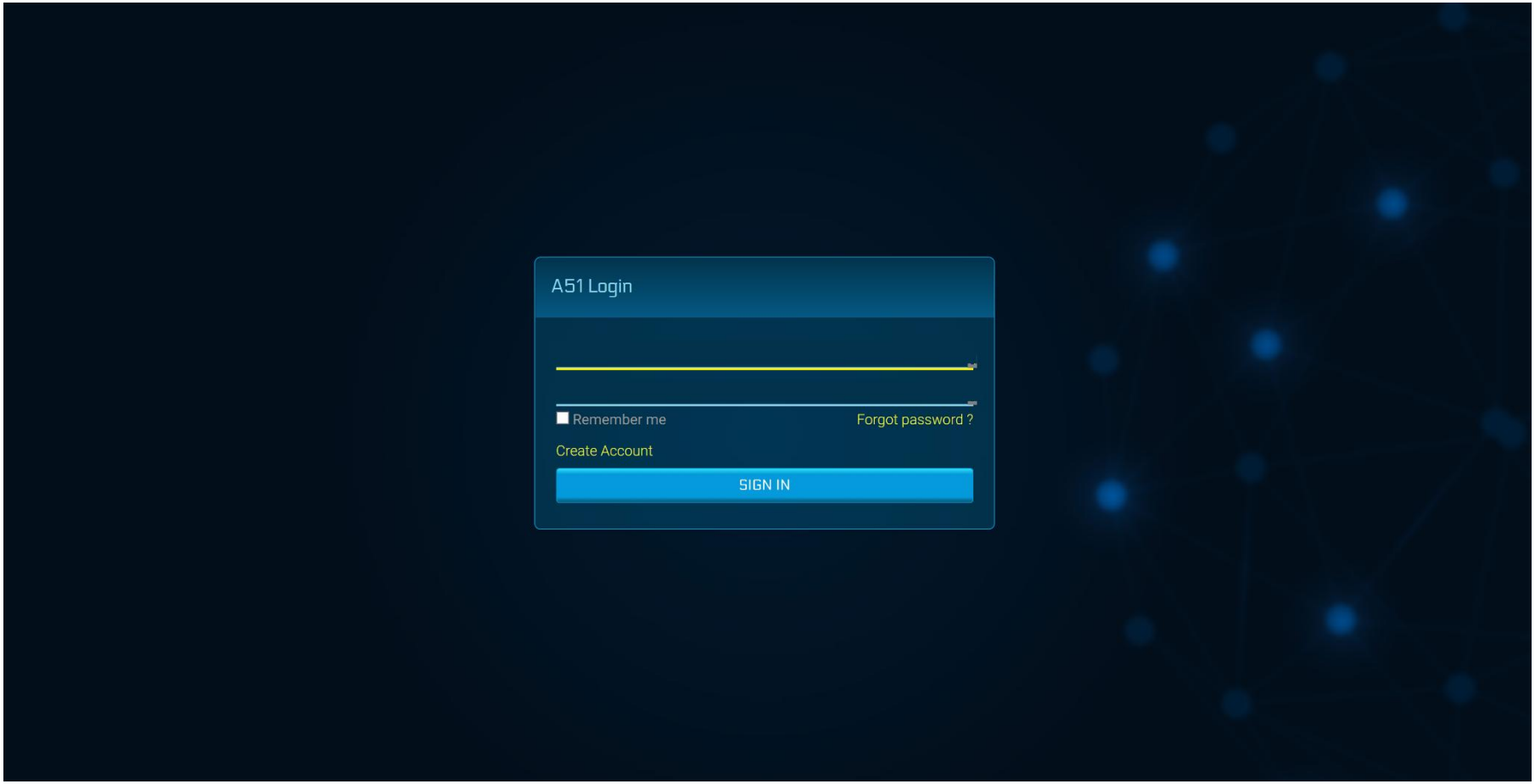
Be sure to check out our main site for useful **TIPS** and assistance.

https://amistacx.io

- Register for A51 [New aMiSTACX Customers must wait 24hrs prior to registration.]
- Read our AWS survival KB article
- **Create a FULL AMI Image/Snapshot backup**
- **Consider updating the Ubuntu System Files and add the latest Security patches.**

Note: Make a full backup first!

XV. A51 Dashboards [Registration]



A51 Monitoring & Control Dashboard will allow a centralized external management of aMiSTACX resources on AWS, and allow you to receive direct support notifications specific to your stack! 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 or did not have a deployment option, 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

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

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