Elegance - Simplicity - Performance - Build <u>your dream</u> Odoo app with **aMiSTACX**

Congratulations!

And welcome to your Premium **Odoo 18 deployment solution by aMiSTACX**

DIY - 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 an **Intermediate** understanding of the AWS console
- You have an **Advanced** 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

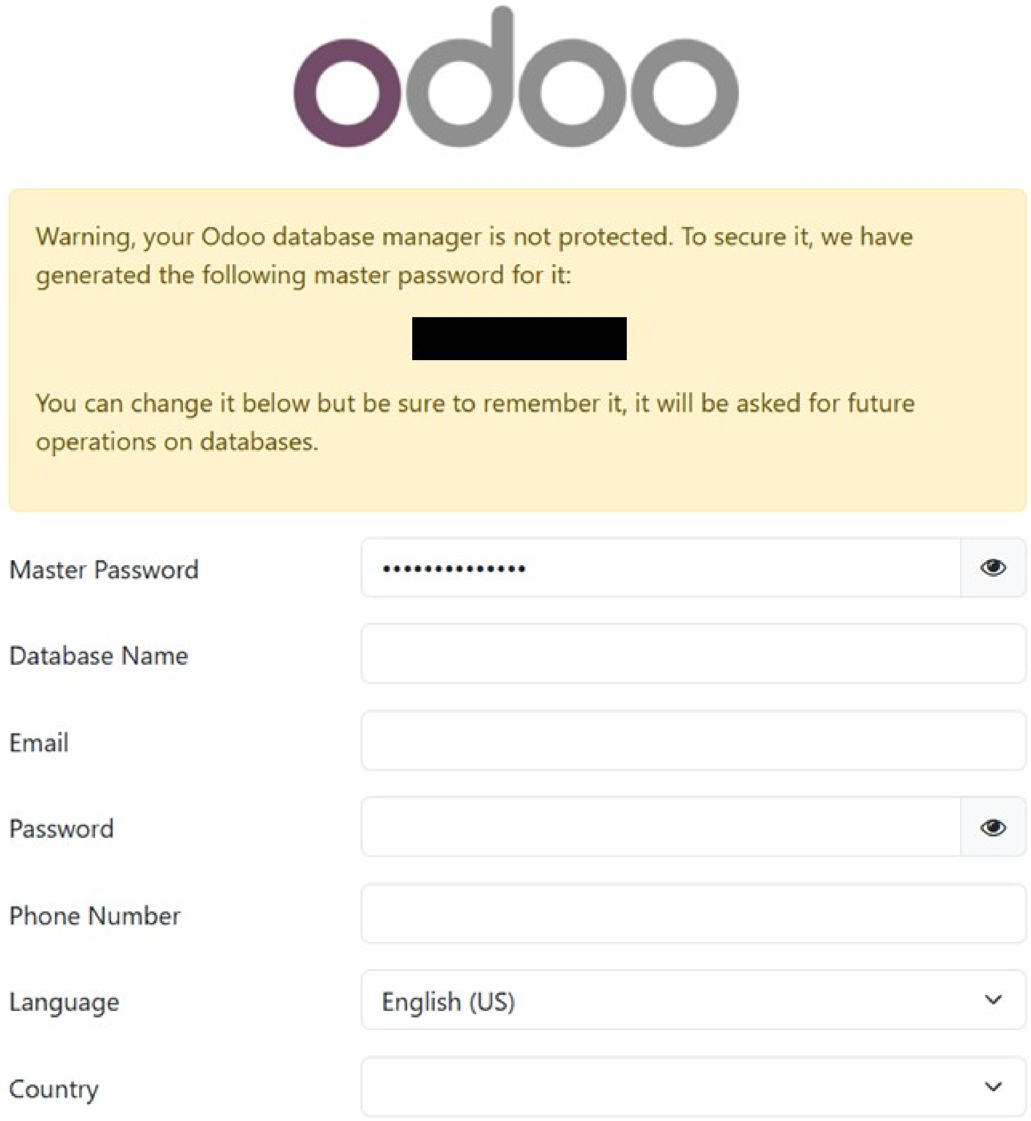
More Info on Putty/AWS: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html

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

Quick Start Guide

After creating the EC2 Instance from the AMI and ensuring that a public IP Address is assigned, and the AWS Security Group allows TCP Ports 22 and 80 inbound to the EC2 Instance, from your web browser, navigate to the Odoo 18 Welcome Screen to start the ODOO database creation for your environment:

http://[Your_EC2_Public_IP_Address]/



Warning, your Odoo database manager is not protected. To secure it, we have generated the following master password for it:

[redacted]

You can change it below but be sure to remember it, it will be asked for future operations on databases.

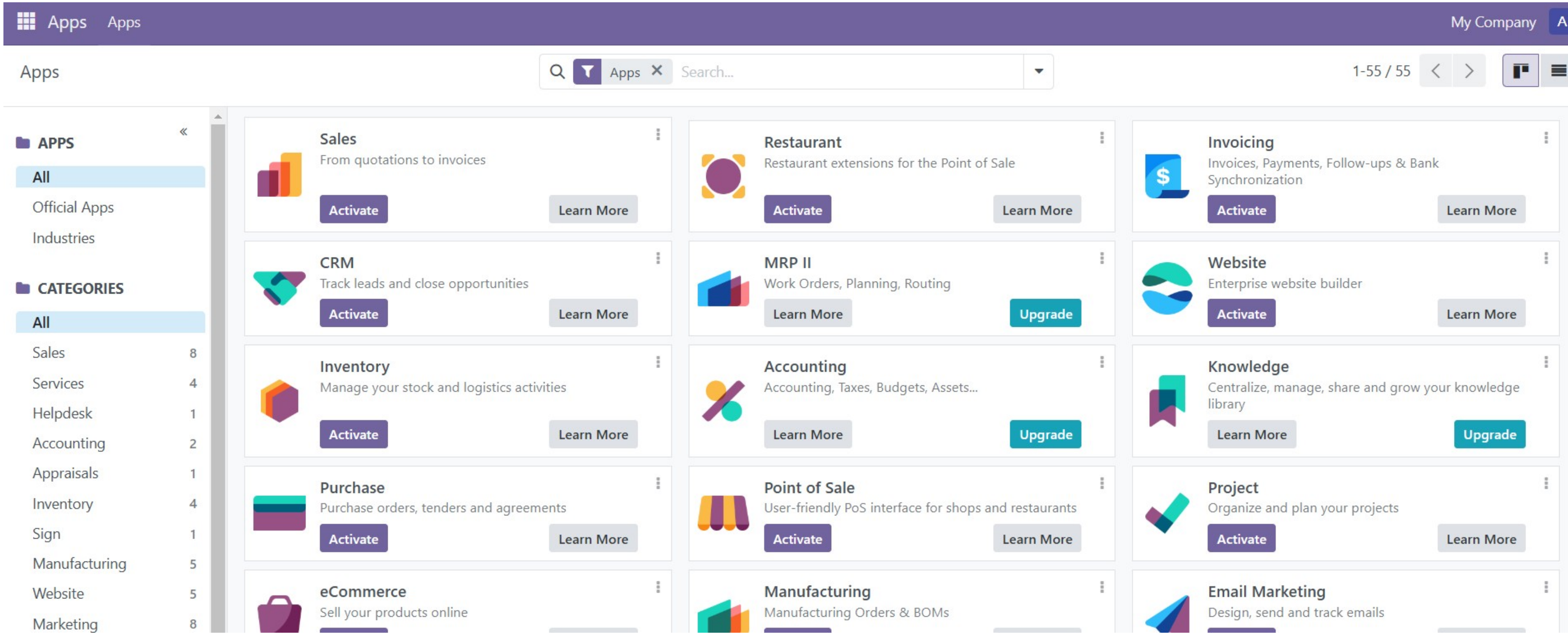
Master Password	•••••••••••••••
Database Name	
Email	
Password	
Phone Number	
Language	English (US)
Country	

Odoo 18 Database Create

- Master Password = **{Your AWS EC2 Instance ID}**
- Database Name = **odoo**

- Email = **{Your email address}**
- Password = **{Your AWS EC2 Instance ID}**
- Master Password = **{Your AWS EC2 Instance ID}**

Once the database has been created, you are ready to customize the application to your requirements.



All	
Sales	8
Services	4
Helpdesk	1
Accounting	2
Appraisals	1
Inventory	4
Sign	1
Manufacturing	5
Website	5
Marketing	8

Sales
From quotations to invoices
Activate Learn More

Restaurant
Restaurant extensions for the Point of Sale
Activate Learn More

Invoicing
Invoices, Payments, Follow-ups & Bank Synchronization
Activate Learn More

CRM
Track leads and close opportunities
Activate Learn More

MRP II
Work Orders, Planning, Routing
Learn More Upgrade

Website
Enterprise website builder
Activate Learn More

Inventory
Manage your stock and logistics activities
Activate Learn More

Accounting
Accounting, Taxes, Budgets, Assets...
Learn More Upgrade

Knowledge
Centralize, manage, share and grow your knowledge library
Learn More Upgrade

Purchase
Purchase orders, tenders and agreements
Activate Learn More

Point of Sale
User-friendly PoS interface for shops and restaurants
Activate Learn More

Project
Organize and plan your projects
Activate Learn More

eCommerce
Sell your products online

Manufacturing
Manufacturing Orders & BOMs

Email Marketing
Design, send and track emails

Odoo Single Site

As this stack is designed for experienced Odoo admins and developers, the ins and outs of advanced Odoo will be reserved for the official Odoo site:

https://www.odoo.com/documentation/18.0/administration/install.html

https://www.odoo.com/documentation/18.0/administration/install/deploy.html

Odoo Paths and Files

/etc/odoo/odoo.conf #Main configuration file

/usr/lib/python3/dist-packages/odoo/ #Main application path

/etc/apache2/sites-enabled/
/etc/nginx/sites-enabled/ #Main vhost for Odoo

/var/log/odoo/ #Basic Log

G-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
```

I. Ubuntu 24.04.2 LTS Essentials

Core Software Versions

- Ubuntu 24.04.2
- Apache 2.4.63
- NGINX 1.28.0
- PostgreSQL 14.18
- Odoo 18 Community
- Wkhtmltopdf 0.12.5

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 EC2 t-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 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

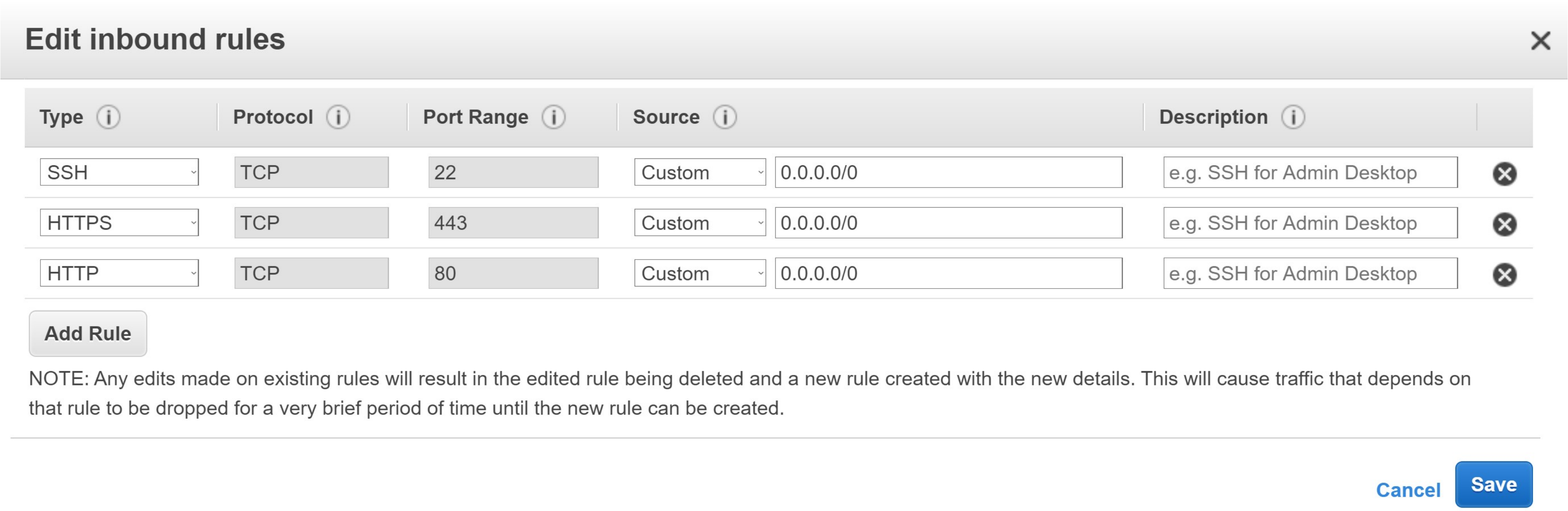
/etc/php/8.x/fpm

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

```
max_execution_time = 300
```

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 8069 [Odoo]



Edit inbound rules

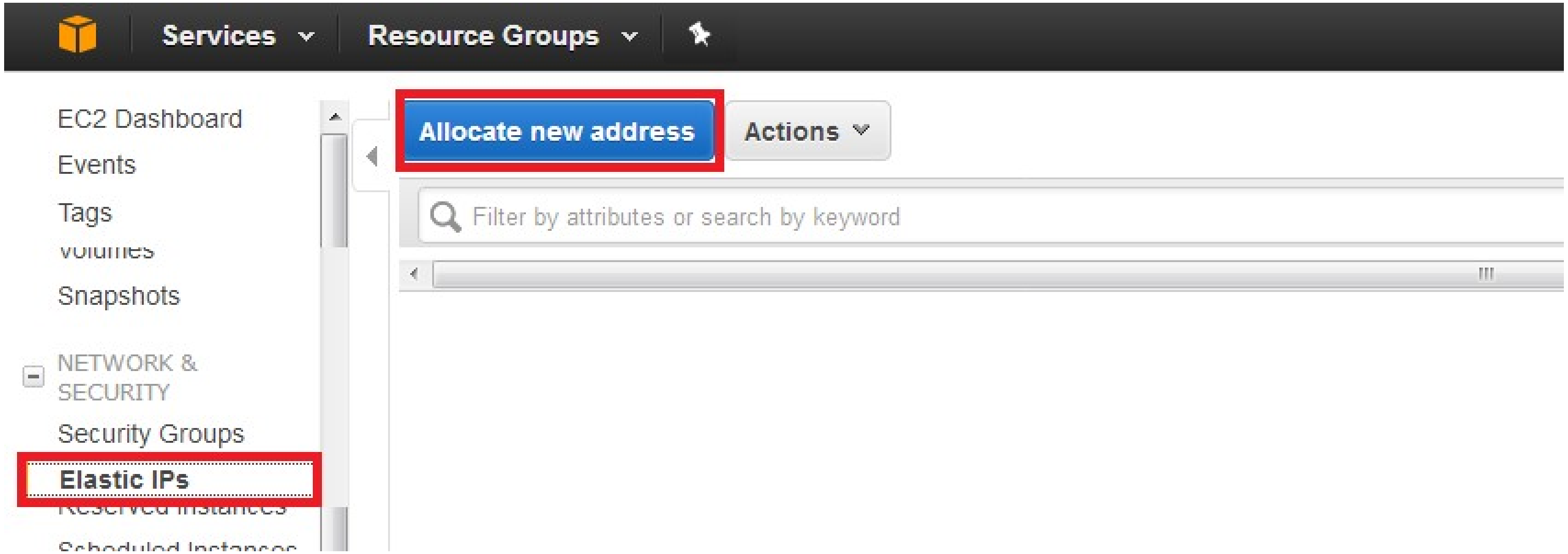
Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

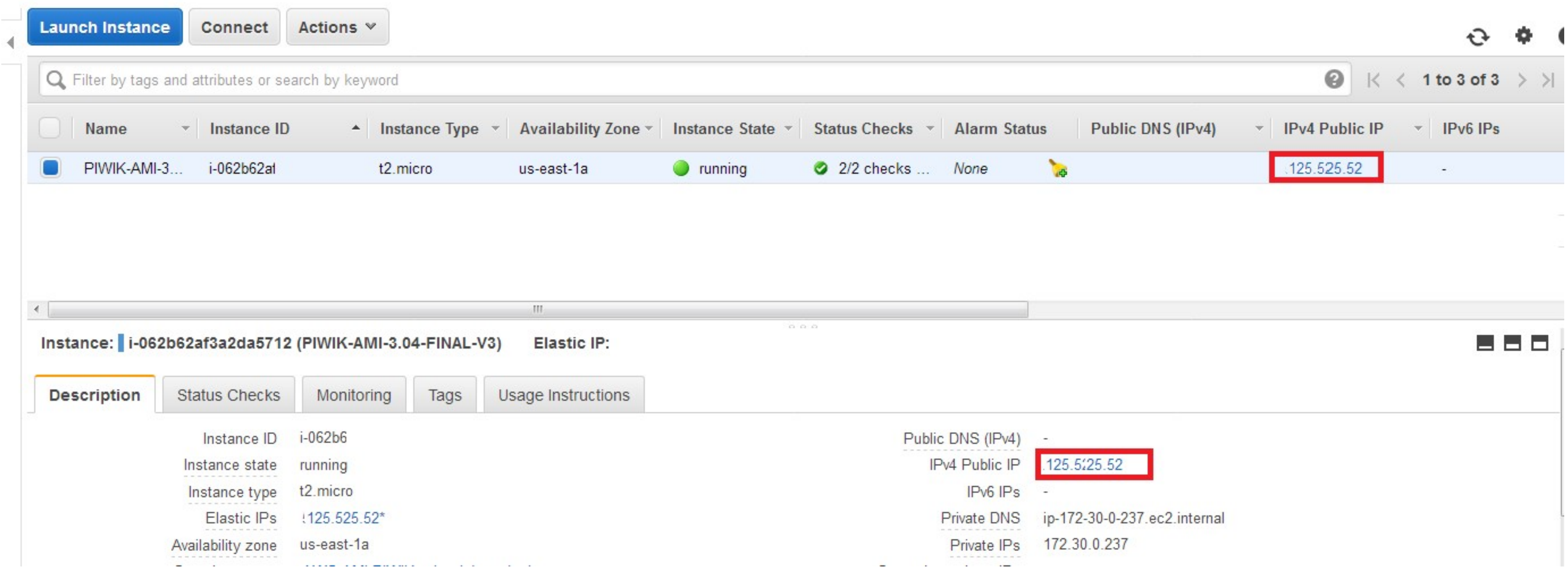
Note: It is recommended that you verify everything is working before changing the SSH to only allow specific connections.

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]



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs
PIWIK-AMI-3...	i-062b62af	t2.micro	us-east-1a	● running	✓ 2/2 checks ...	None		.125.525.52	-

Instance: i-062b62af3a2da5712 (PIWIK-AMI-3.04-FINAL-V3) **Elastic IP:**

Instance ID	i-062b6	Public DNS (IPv4)	-
Instance state	running	IPv4 Public IP	125.5:25.52
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs	:125.525.52*	Private DNS	ip-172-30-0-237.ec2.internal
Availability zone	us-east-1a	Private IPs	172.30.0.237

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>/**

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: Area 51 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/odoo.conf

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

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

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

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

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

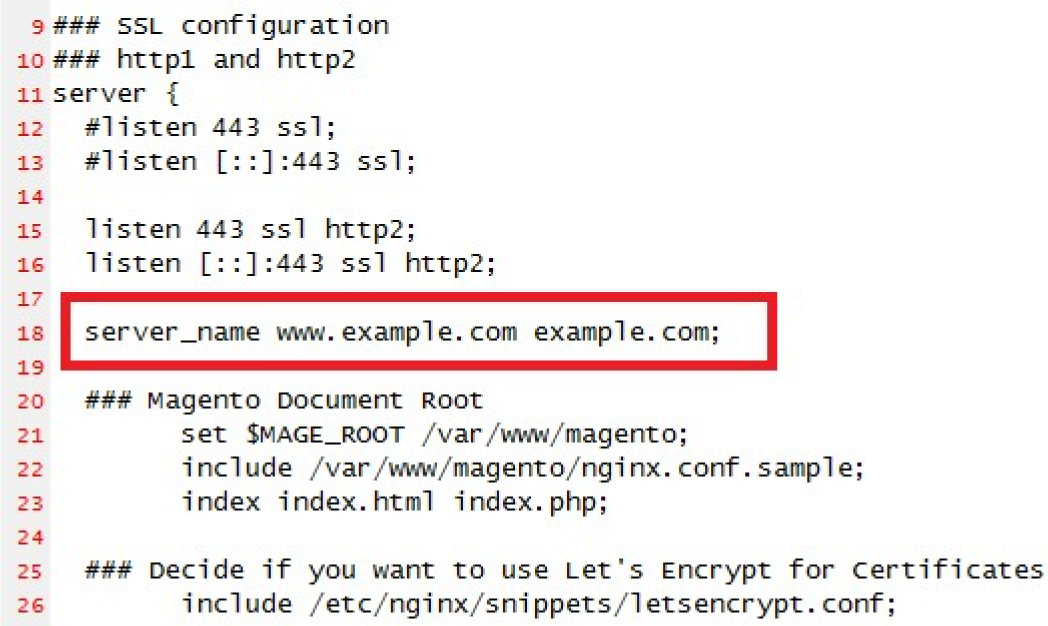
Save files! And from from CLI: **sudo service apache2 restart**

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/odoo



```
### 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 >>

Apache Reverse Proxy and HTTPS

To set up a FQDN and HTTPS is very easy. [Video on our Y/T Channel]

- First add your server name to the apache vhost files and then restart the service.
- You MUST have a cert. Do not use the self-signed.
- If you are using Cloudflare, and you should, then turn CDN off and get a cert from Let's Encrypt. [See Let's Encrypt Section]
  [You really don't need a cert when using Cloudflare, but it is good practice anyway.]
  Note: Don't let select option 2 for LE redirect! Do it manually.
- Once you have the local cert, using the correct vhost format, you will not need to use a port number. Your URL will resolve in the following format for Odoo: **https://mydomain/web/login ;** You can even have multiple sites as long as you have unique domain names and port as defined in the vhost files.
- Set host file reverse proxy DNS for EC2.

Note: It's best to make use of Cloudflare's origin certificates as they do not need to be renewed every three months.

NGINX Reverse Proxy and HTTPS [Advanced]

Note: This is not a step by step, it is for advanced users that understand how to work with NGINX in a reverse proxy. It is only a general overview.

- Disable the vhost odoo.conf and enable the reverse proxy conf. [Call it what you want.]
- Add your server domain name to the nginx vhost file [remove example.com] and then restart the nginx service.
- If you are using Cloudflare, and you should, then turn CDN off and get a cert from Let's Encrypt. [See Let's Encrypt Section]
  [You really don't need a cert when using Cloudflare, but it is good practice anyway.]
  Note: Don't let select option 2 for LE redirect! Do it manually. The vhost reverse-proxy the templates you want.
- Set /etc/odoo12.conf to Proxy mode, and restart Odoo service.
- Once you have the local cert, using the correct vhost format, and you are in proxy mode, you will not need to use a port number and you can use HTTPS. Your URL will resolve in the following format for Odoo: **https://mydomain/web/login ;** You can even have multiple sites as long as you have unique domain names and port as defined in the vhost files.

Odoo Enterprise

We have clients using Odoo Enterprise on this stack without issue.

Basically, you create an enterprise folder within Odoo in the addons.

E.g. **/usr/lib/python3/dist-packages/odoo/addons/enterprise**

- You add the Enterprise modules
- You update the configuration file as shown in the odoo18.conf.multi for the path statement
- You restart the Odoo service
- You sign into odoo and switch to developer mode
- You install the Enterprise Module
- You run update
- You add your Enterprise Key

Misc

To restart the Odoo service:

```
sudo systemctl restart odoo
```

Odoo Basic Security

- The PostgreSQL User = **odoo**
- The psql user odoo password = **{Your AWS EC2 Instance ID}**
- Master Password = **{Your AWS EC2 Instance ID}**
- Postgres user’s password = **{Your AWS EC2 Instance ID}**

Important! It is recommended you change all of the passwords to unique values.

PostgreSQL

Ref: https://www.postgresql.org/

```
/usr/bin/psql -V
```

Reset postgres passwords

```
sudo -u postgres psql -c "ALTER USER odoo PASSWORD '[new_password]';"
sudo -u postgres psql -c "ALTER USER postgres PASSWORD '[new_password]';"
```

Odoo Performance

Consult the Odoo official guide:

https://www.odoo.com/documentation/18.0/administration/install/deploy.html

/etc/odoo/odoo.conf #Main configuration file

AWS Instance Types Ref: https://aws.amazon.com/ec2/instance-types/

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 scripts in /var/www/utility

XIII. Post Install Security Considerations

1. SSH Security Group

Consider restricting access to the SSH and 8080 port via your AWS security group. As per the below article outlines.

https://amistacx.io/restrict-access-to-ssh-with-aws-security-groups

2. Lock down Odoo ports via AWS Security Groups, or bind to 127.0.0.1

3. Review post deployment suggestions

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

XIV. What’s Next?

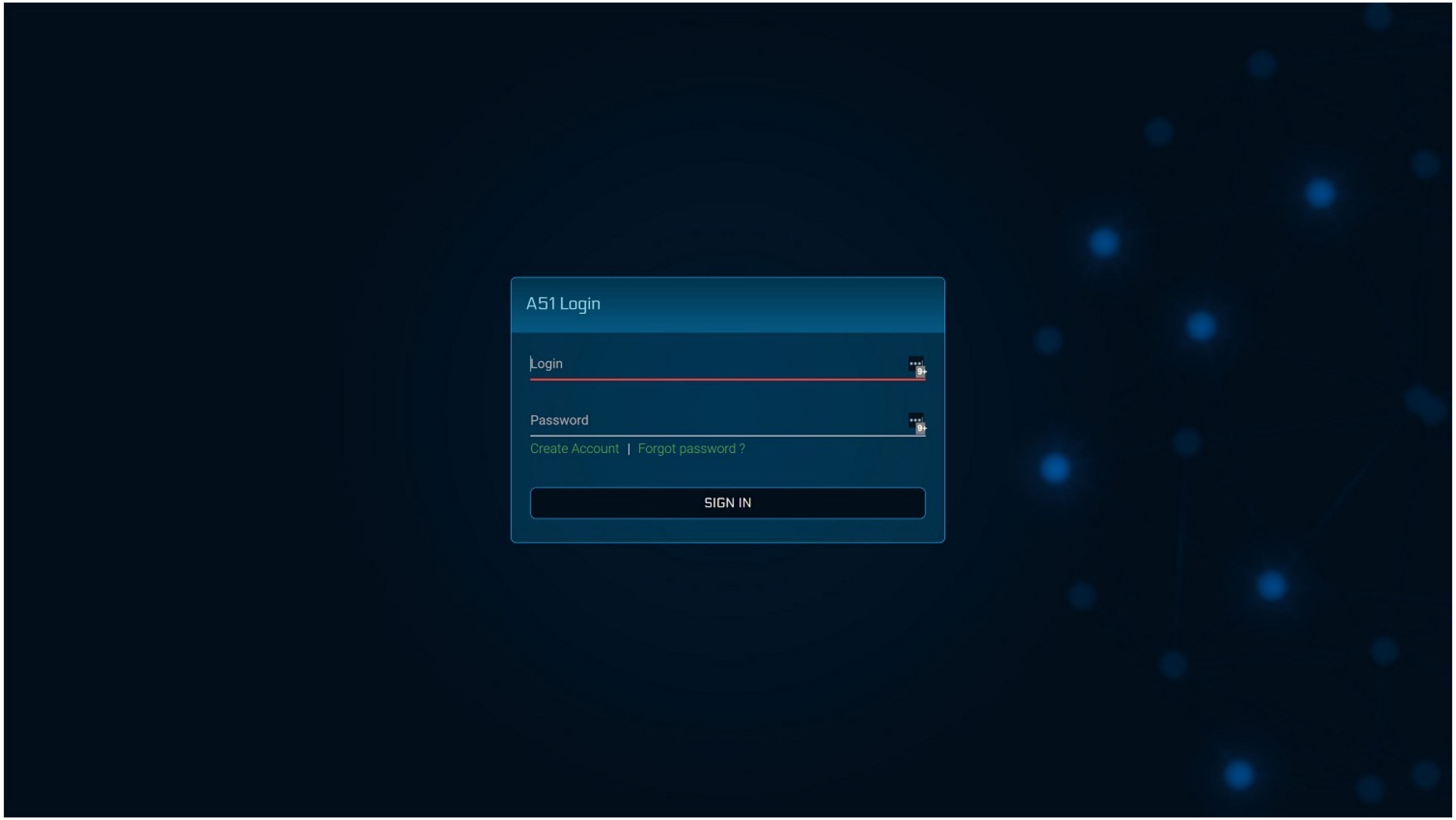
Be sure to check out our main site for useful **TIPS** and assistance.

https://amistacx.io

- Register on A51
- Create a FULL AMI Image/Snapshot backup
- Consider updating the Ubuntu System Files and add the latest Security patches.

https://amistacx.io/how-to-patch-the-ubuntu-os

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

Every aMiSTACX stack now is preconfigured with the AWS CloudWatch Agent. There are also advanced features included that interface with the A51 Dashboard.

If you wish to use A51 Monitoring, please enable the **collectd** service. Keep in perspective this process will consume additional server resources.

```
sudo systemctl enable collectd
```

```
sudo systemctl start collectd
```

More info:

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

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

XVI. 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!**