Installation Guide

Monitor Lizard is a Django project written in Python 3. Monitor Lizard consists of several separable components, all of which may be run on the same or different servers, depending on the scale and scope of your deployment. These components are:

|  |  |  |
| --- | --- | --- |
| Component | Purpose | Connects To (use this to plan your firewalls): |
| Database | Stores persistent application data | * Web Server | SQL * Processing Daemon | SQL * Alert Daemon | SQL |
| Web Dashboard | Serves administration | * Database | SQL * Host Daemon | HTTPS |
| Message Bus | Carries messages from Hosts to Processing Daemon | * Host Daemon | AMQP * Processing Daemon | AMQP |
| Processing Daemon | Stores reports in Database | * Message Bus | AMQP * Database | SQL |
| Host Daemon\* | Reports data about hosts | * Web Server | HTTPS * Message Bus | AMQP |
| Alert Daemon | Notifies users about alerts | * Database | SQL |

\* You will run the Host Daemon on every host that reports to the system

This installation guide will walk you through setting up each component.

## Database

The preferred database is PostgreSQL, but you may use any of the following databases supported by Django: PostgreSQL, MariaDB, MySQL, Oracle, SQLite. Set up your database as normal, and note the credentials and hostname/ip address.

Note: If you are not concerned with performance **and** plan to run all of your components from the same MonitorLizard folder, you may skip this step and use the default SQLite database.

## Web Dashboard

1. Copy the latest release of Monitor Lizard onto your web server host.
2. Make sure you have installed Python 3 and the following python libraries:
   1. django
   2. psutil
3. Configure Monitor Lizard to use your database. Place the credentials and hostname/ip address of your database in MonitorLizard/MonitorLizard/settings/components/base.py (the file has links to documentation).
4. You may use WSGI or ASGI to deploy the web dashboard (for more information, [refer to the relevant django documentation](https://docs.djangoproject.com/en/3.2/howto/deployment/)).

Note: If you are not concerned with performance or security, you may use [djangos runserver command](https://docs.djangoproject.com/en/3.2/ref/django-admin/#runserver) instead of a web server.

1. Once your dashboard is deployed, run *python manage.py setup*.
   1. Agree to flush the database (this won’t do anything scary).
   2. Set an email and password for the *admin* superuser.
   3. Confirm you can log into *example.com/yoururl/* and *example.com/yoururl/admin/*.

## Message Bus

Monitor Lizard was developed with RabbitMQ, but it should work with any AMQP-compatible message queuing software. We found the following docker command worked for small scale deployments:

docker run -it --rm --name rabbitmq -p 5672:5672 -p 15672:15672 rabbitmq:3-management

Install your message queue of choice, and note the ip address.

## Processing Daemon

1. Copy the latest release of Monitor Lizard to a server that can access the Database and Message Bus components.
2. Make sure you have installed Python 3 and the following python libraries:
   1. django
   2. pika
3. Change the ip address in MonitorLizard/web/message\_queue.py.MessageQueue.\_\_init\_\_ to the ip address of your Message Bus.
4. Configure Monitor Lizard to use your database. Place the credentials and hostname/ip address of your database in MonitorLizard/MonitorLizard/settings/components/base.py (the file has links to documentation).
5. Run python manage.py processing\_daemon &
6. For a more robust solution for permanently running the Processing Daemon, we recommend researching [supervisord](http://supervisord.org/) or other supervisory programs. Restarting daemons after exceptions or power cycles is not within the scope of Monitor Lizard.

## Host Daemon

Do the following for every host you want reporting data to Monitor Lizard:

1. Copy the latest release of Monitor Lizard to the host you want to monitor.
2. Make sure you have installed Python 3 and the following python libraries:
   1. django
   2. psutil
   3. pika
3. Change the ip address in MonitorLizard/web/message\_queue.py.MessageQueue.\_\_init\_\_ to the ip address of your Message Bus
4. Run python manage.py host\_daemon &
5. For a more robust solution for permanently running the Host Daemon, we recommend researching [supervisord](http://supervisord.org/) or other supervisory programs. Restarting daemons after exceptions or power cycles is not within the scope of Monitor Lizard.

## Alert Daemon

1. Copy the latest release of Monitor Lizard to a server that can access the Database component.
2. Make sure you have installed Python 3 and the following python libraries:
   1. Django
   2. django-split-settings
3. Configure Monitor Lizard to use your database. Place the credentials and hostname/ip address of your database in MonitorLizard/MonitorLizard/settings/components/base.py (the file has links to documentation).
4. Override the default email credentials in settings by placing the EMAIL\_HOST, EMAIL\_USE\_TLS, EMAIL\_PORT, EMAIL\_HOST\_USER, EMAIL\_HOST\_PASSWORD in MonitorLizard/MonitorLizard/settings/components/email.py (create the file manually under the directory)
5. Run python manage.py alert\_daemon &
6. For a more robust solution for permanently running the Alert Daemon, we recommend researching [supervisord](http://supervisord.org/) or other supervisory programs. Restarting daemons after exceptions or power cycles is not within the scope of Monitor Lizard.

When the product is delivered to the sponsor, build, installation and configuration steps will typically be required. Unless otherwise specified by the sponsors, the Installation Guide should describe the process of deploying the system to a production environment rather than a developer setup. Assume that developers or IT operations staff are the target audience. The Installation Guide should also contain instructions for verifying that the system installation is working properly after installation. The Guide should include details about how to configure the system or add the first users to the system (if appropriate). Be sure that the installation does not include test or dummy data. For a web application, provide instructions for installing in a production web server, as opposed to running a development web server from source from a local checkout. This includes providing guidelines and suggestions to improve the security of a deployment (e.g., running the application exclusively under HTTPS).

The Installation Guide must address the following:

Identification of all tasks required for product build and installation, by providing specific examples of commands to run, including installation of all dependencies. Include brief explanations of what’s being done if it might be non-obvious. If the system requires a database, this guide should provide clear instructions for where and how to configure connection parameters. Include all steps to install the application in a production installation, suitable for inclusion in a shell script or entering via a command-line tool, if possible. Automating the install will allow for fewer steps to be documented.

A troubleshooting guide that describes the steps that will be taken for common problems that may be encountered during build or installation.

A verification plan that describes the means to be used to verify that the project installation is successful. This might include checking program output, sending web requests, and so forth.