MONARC technical guide

"security made in Lëtzebuerg" (SMILE) g.i.e.

Table of Contents

1. Introduction

This document is currently under active development and discussion!

This document is intended to administrators of a MONARC instance. If you find errors or omissions in this document, please don't hesitate to submit an issue or open a pull request with a fix.

2. Architecture

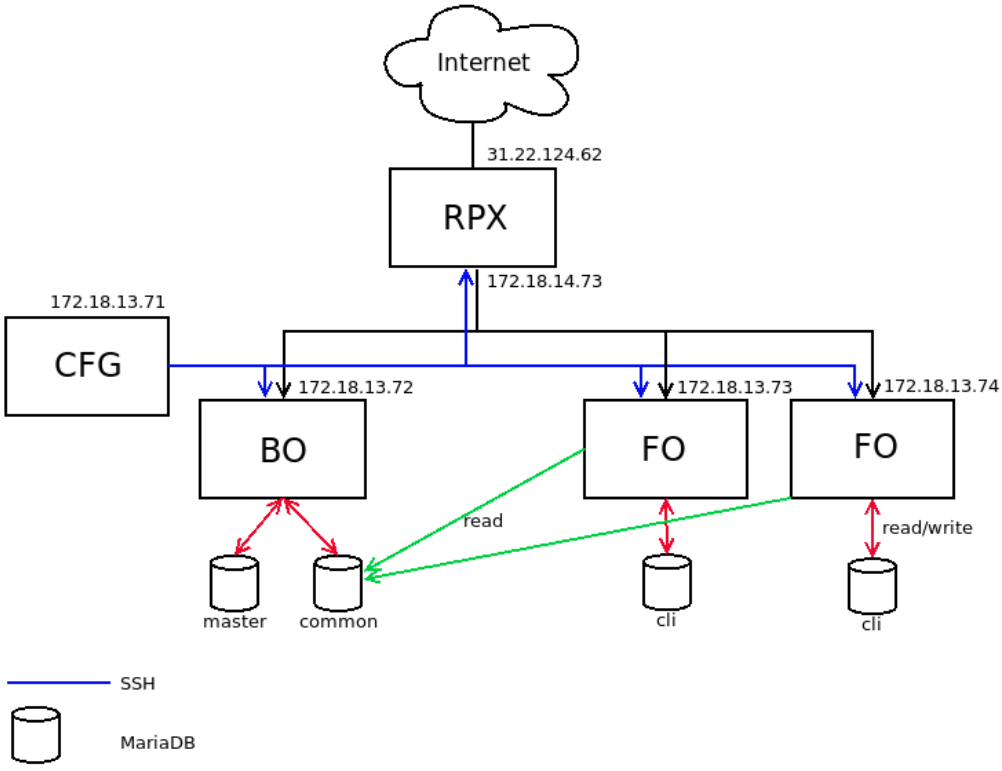
2.1. Global architecture

Basically MONARC is composed of two main parts:

- a back office: for the management of the database, servers and clients of MONARC installations;
- a front office: for the management of the risks analysis. A front office instance can be used without the back office.

If you want run your own risk analysis you only need the front office. More information about the installation in this section.

The whole architecture includes an additional reverse proxy and a configuration server. It is possible to connect many front offices to one back office.



It can be deployed with ansible. More information here.

- 172.18.14.73 - monarc2-rpx.test.your.domain.com

- 172.18.13.71 - monarc2-conf.test.your.domain.com
- 172.18.13.72 - monarc2-master.test.your.domain.com
- 172.18.13.73 - monarc2-fo01.test.your.domain.com
- 172.18.13.74 - monarc2-fo02.test.your.domain.com

2.1.1. Network requirements

The configuration server (CFG) manages the configurations of the back office, reverse proxy and different front office(s) via SSH.

The reverse proxy (RPX) should of course be available from Internet.

The database *common* of the BO should also be available to the FO servers.

Postfix should be installed on the BO and on the FO. Indeed, SMTP is used for the account creation and the password recovery. The easiest configuration is to set up Postfix for relaying emails through your mail server.

2.2. Specifications by server

The specifications below are of course only advices. It will vary depending on your need.

Back office:

- 4 vCPU
- 16GB RAM
- 25GB HDD
- Ubuntu Server 16.04.2 LTS 64 bits

Front office:

- 4vCPU
- 16GB RAM
- 25GB HDD
- Ubuntu Server 16.04.2 LTS 64 bits

Reverse proxy:

- Not that powerful, but all connections (for the BO or FO) will go through it. The bandwidth must be good.
- Ubuntu Server 16.04.2 LTS 64 bits

Configuration server:

- Not that powerful. Mainly used for the creation of clients on the different FOs.
- Ubuntu Server 16.04.2 LTS 64 bits

- We have tested with version 2.2.1.0 of ansible.

3. Modules of the project

The source code of MONARC is divided in several modules with dedicated Git repositories.

Summary of the different modules of the project

	Back office	**Front office**
	MONARC BO: Management of the database, clients and FO servers.	MONARC: The MONARC front office
Modules	• MonarcCore (zm-core in `vendor/monarc/core`) • MonarcBO (zm-backoffice in `vendor/monarc/backoffice`)	• MonarcCore (zm-core in `vendor/monarc/core`) • MonarcFO (zm-client in `vendor/monarc/frontoffice`)
Interfaces (in node_modules/)	• ng_anr • ng_backoffice	• ng_anr • ng_client

- MonarcCore (zm-core) is the API in charge of providing an access to the data. Common to the front office and the back office.
- MonarcFO (zm-client) is the interface for the front office.
- MonarcBO (zm-backoffice) is the interface for the back office.

4. Requirements and deployment

4.1. Requirements

The deployment of MONARC has been tested on Debian 9.0, Ubuntu 16.04 LTS and Ubuntu 17.04.

- PHP (version 7.0 recommended);
  - PHP extensions: xml, mbstring, mysql, zip, unzip, mcrypt, intl, gettext, imagick
- Apache 2;
- MariaDB;
- Postfix (SMTP is used for the account creation and password recovery).

4.2. Deployment

4.2.1. Only the front office

Different methods are provided.

Prepared virtual machine

A virtual machine for use with VirtualBox is available if you want to quickly test MONARC.

The best way if you want to try an up-to-date version of MONARC.

Vagrant

If you want to deploy the application with Vagrant:

```
$ sudo apt-get install virtualbox vagrant
$ git clone https://github.com/monarc-project/MonarcAppFO-vagrant.git
$ cd MonarcAppFO-vagrant/
$ vagrant up
```

Once the virtual machine will be ready go to the address: http://127.0.0.1:5001
The username is *admin@admin.test* and the password is *admin*.

This method is not recommended for production.

Manual deployment

Follow the instructions here.

4.2.2. The back office and the front office

The whole architecture can be deployed with ansible. More information here.

5. Updates

Before updating MONARC it is strongly advised to configure database backup. For that you just need to create a file `data/backup/credentialsmysql.cnf`:

```
[client]
host     = localhost
user     = sql-monarc-user
password = your-password
socket   = /var/run/mysqld/mysqld.sock
[mysql_upgrade]
host     = localhost
user     = sql-monarc-user
password = your-password
socket   = /var/run/mysqld/mysqld.sock
basedir  = /usr
```

If this file is not present, a warning message will be displayed during the update.

5.1. System update

Keep the software of your distribution up-to-date (Apache, PHP, MariaDB, etc.). At least the security updates from the GNU/Linux distribution.

5.2. MONARC update

The master branch should always be working and it is recommended to install the project using this one.

If you have already installed MONARC and want to update to a later version, you can use the provided script:

```
$ scripts/update-all.sh
$ sudo systemctl restart apache2
```

This script will retrieve the updates from the last stable release of MONARC.

For more details the script will:

- backup your databases in case the update fails;
- check the presence of composer;
- retrieve the new code from the Git repositories from the last stable release:
  - pull updates for module/[MonarcCore, MonarcBO/MonarcFO];
  - pull updates for node_modules/[ng_anr, ng_backoffice/ng_client].
- run the appropriate database upgrade scripts (pathCore, pathBO, pathFO);
- install/update JavaScript libraries with npm;
- update the translations.