AGNITAS
email & marketing automation

Install & Admin Guide

for

OpenEMM 22.10

(22.10.000)

AGNITAS AG

www.openemm.org

forum.openemm.org

Author:
Martin Aschoff

Table of Contents

© 2023 AGNITAS AG

© 2023 AGNITAS AG

© 2023 AGNITAS AG

1 Document History and Outlook

Version	Date	Change
1.0.0	December 9, 2022	Section 3.1: Updated software stack and provided outlook Section 4.1: Added "msgpack" and "websockets" to list of Python packages Section 5.1: Changed MariaDB version from 10.5.8 to 10.6.8 and recommended an important bug fix Section 5.1+9.3: Added option to specify version of Python module "mariadb" Section 9.6: Instructions for upgrading 22.04 to 22.10 Section 11.2: New section on monitoring of OpenEMM's jobqueue Section 12.1.1: Replaced APR with NIO Connector, because APR has become deprecated
	December 9, 2022	**Document fork for OpenEMM 22.10**
1.0.0	September 15, 2022	Some minor additions derived from the EMM Install&Admin Guide
	March 8, 2022	**Document fork for OpenEMM 22.04**
1.0.6	February 22, 2022	Some editorial adjustments
1.0.5	November 26, 2021	Chapter 4: Moved Python chapter to better reflect the required setup sequence Section 5.1: Extended instructions for deployment of MariaDB Section 10.1: Enhanced explanations of options to configure bounce management
1.0.4	November 23, 2021	Section 3.4: Extended instructions to enable log file of MariaDB
1.0.3	November 19, 2021	Section 4.5: revised explanation for journald and rsyslog changes Section 10.1.1: added hint to forward port 443 to 8443
1.0.2	November 14, 2021	Section 3.4: Extended instructions on how to install your favourite version of MariaDB from archive.mariadb.org
1.0.1	November 1, 2021	Introduced cross-references for easier navigation within this guide Requirement of Python 3.8 (or later) emphasized in various places Section 9.1: Added LimitNOFILE value to script for Tomcat startup
1.0.0	September 30, 2021	Section 3.1: Added AlmaLinux as replacement for CentOS 8 Section 3.1+3.1.1: Update of supported software stack and its roadmap Section 3.4: Changed source for MariaDB from repo to archive.mariadb.org Section 7.4: Added special advice for upgrading to version 21.10 (required: Java 11 + Tomcat 10) and added info on how to migrate a server from CentOS to AlmaLinux Removed all Sendmail references, because Sendmail is no longer supported
	September 30, 2021	**Document fork for OpenEMM 21.10**

© 2023 AGNITAS AG

2 Introduction

2.1 Purpose of OpenEMM

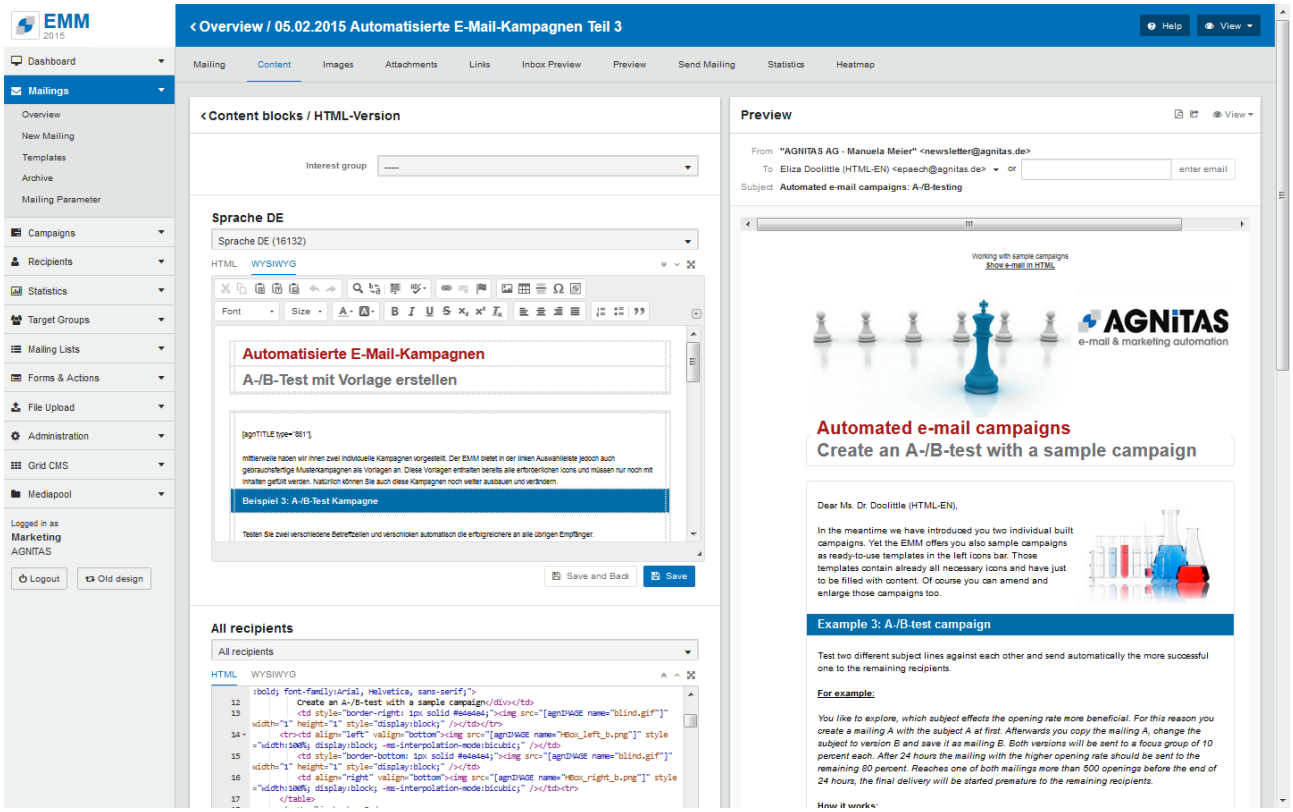
OpenEMM is a web-based enterprise application for email marketing, email newsletters, service emails (transaction emails and event or time triggered emails), marketing automation and lead management. To summarize it, OpenEMM is a tool for customer relationship management by email.

EMM offers tons of features for professional marketing users, among them:

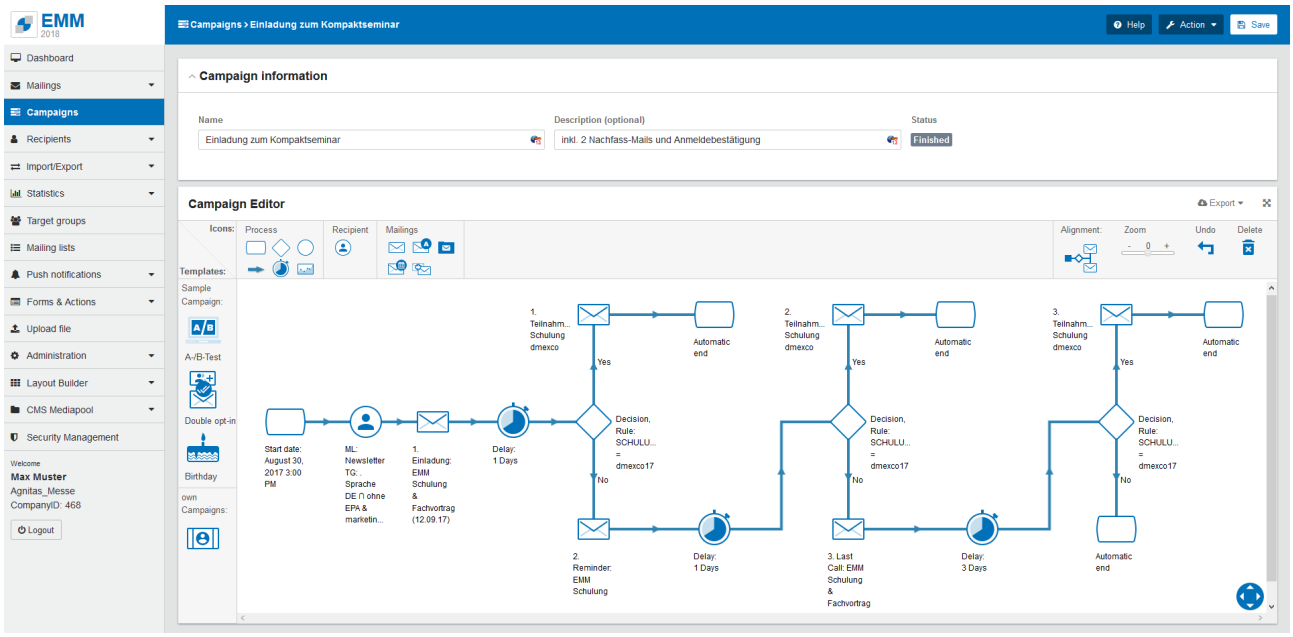
- a console based administration tool for checks, configuration, updates and backups
- a responsive web user interface with great usability and different languages
- a mail template management system
- a visual web form builder
- mailing, template and web forms import and export to load and exchange prepared mailing templates and web forms
- ability to fill mail templates with data from databases and realtime content created on-the-fly
- a graphical workflow manager to create complex campaigns with drag&drop
- individual and (GDPR compliant) anonymous tracking of mail openings, link clicks and deep tracking
- automated bounce management
- graphical realtime statistics with lots of KPIs
- self-defined target groups based on recipient profiles and recipient's behaviour (created visually or with SQL-like syntax)
- a Postfix based mail backend for maximum sending performance with DKIM support
- flexibly configurable data import and export with extensive reporting of results
- sophisticated management of users, user roles and user rights
- an audit-proof searchable and exportable user activity log
- a system status menu with helpful info and configuration options for OpenEMM administrators
- a scripting feature to enhance the functionality of EMM with triggerable customized actions
- an extensive set of SOAP webservices to manage OpenEMM from remote
- a universal RESTful API to access OpenEMM from remote with HTTP(S) links

The GUI of OpenEMM works with web browsers Google Chrome, Mozilla Firefox and Microsoft Edge. To give you an impression of the web-based user interface of OpenEMM, the following two screenshots illustrate how to build mailings and how to build workflows and campaigns with OpenEMM:

© 2023 AGNITAS AG



Mailing creation with WYSIWYG editor (upper left) HTML editor (lower left) and preview (right)



Building of workflows and campaigns with OpenEMM's graphical drag&drop editor

© 2023 AGNITAS AG

2.2 General Architecture

OpenEMM consists of several independent services for scalability. It runs on top of a well proven Open Source software stack without any further dependencies to any commercial software:

- Red Hat Enterprise Linux (**RHEL**) or Suse Linux Enterprise Server (**SLES**) or compatible
- Postfix
- MariaDB (or MySQL 5.7)
- Java OpenJDK 11
- Apache Tomcat 10
- Python 3

2.3 Open Platform Design

A software like OpenEMM must not operate as an island, but it should be capable to be integrated with third party systems like a CRM or CMS software, an e-commerce shop, an ERP platform or a business intelligence software. Therefore, OpenEMM was designed to be a very open and flexible platform with lots of interfaces and extension capabilities. It can be used headless, too.

OpenEMM offers not only a highly customizable graphical interface (GUI) for its users, but also an easy to use URL API für URI tunneling and a rich SOAP webservice interface as well as a RESTful API to be used by third-party software.

2.4 Architecture Components

The OpenEMM software is not a monolith, but it is divided into several independent backend and frontend **services** (software). Communication between the various OpenEMM services is done via the database, i.e. the database is the hub of the application.

To gain a better understanding of the purpose of the various backend and frontend services, this is a summary of their main jobs:

- **Backend:** The backend services read mail-related data from the database, generate the individual emails, create mail previews and send the mails out into the Internet. Backend services also register instant bounce messages and collects reponses (like autoresponder and mail replies by recipients) and delayed bounce data to update the database with results from the mail sendings. Backend services send out autoresponder mails defined in the OpenEMM GUI and forward all mail replies not filtered out to a predefined destination (feedback address).

© 2023 AGNITAS AG

- **GUI** (part of frontend): This service provides the browser-based user interface of OpenEMM.
- **Statistics** (part of frontend): This service generates statistics (tables with numerical values as well as visual charts and diagrams) for the OpenEMM GUI and creates reports in PDF and CSV format for download.
- **Webservices** (part of frontend): This service provides the webservice interface of OpenEMM.

© 2023 AGNITAS AG

3 General Requirements

This document will guide you through some necessary steps, which are needed to install and configure OpenEMM. It requires a little knowledge of Linux system administration, of MariaDB or MySQL database administration and knowledge of Domain Name Services (DNS) to configure the domain names for various services.

For questions, comments, suggestions for improvement and other contributions to improve this guide, please feel free to use the OpenEMM support forum at forum.openemm.org

3.1 Software Stack

This is the software stack required by OpenEMM **22.10**:

- 64-bit version of **RHEL 7** or **8** or compatible distribution like AlmaLinux, or **SLES 15**
- **Java Open JDK 11** (LTS version)
- Apache **Tomcat 10.0** (required because of Jarkata EE support)
- **Python 3.8** or later (see chapter *Python 3.8 (or later) for OpenEMM* for details)
- DBMS: **MariaDB 10.3** to **10.6** (MySQL 5.7)
- MTA: **Postfix** 2.6 or later (see section *Switching SMTP Server from Sendmail to Postfix* in case you still use Sendmail)
- for image and PDF creation: command line tool **wkhtmltox** 0.12.5 or later

It may be that OpenEMM also operates on later versions than the ones listed above, but this has not been tested in a production environment, yet.

To install OpenEMM and the required software stack you need a shell access (like *bash*) for your server as user *root*. All command-line examples are based on RHEL or SLES. Unless otherwise noted, you should run all commands as user *root* to make sure you own and you can grant the required permissions.

Instructions that are valid only for RHEL, are introduced with a header line **RHEL** and are ended with a blank line. Instructions that are valid only for SLES, are introduced with a header line **SLES** and are ended with a blank line, too.

3.1.1 Software Stack Roadmap / End of Life

Hera are some additional information on the software stack requirement of future versions of OpenEMM, so that you have enough lead time for your release plannings:

- **Linux:** CentOS 8 was discontinued at the end of 2021. We recommend to switch to AlmaLinux 8 instead. See section *Special Advice for Upgrading to 21.10* for details.
- **Java:** OpenEMM is compiled with Java 11 and needs a deployment of Java JDK 11 as runtime environment. For the future, OpenEMM will only support LTS releases of Java JDK, which means, that in the future, it will support Java JDK 17.
- **Tomcat:** Because OpenEMM is compiled with Java 11, it requires Jarkata EE libs. Jarkata EE is only supported by version 10.0 of web container Tomcat.

© 2023 AGNITAS AG

- **MariaDB:** Since OpenEMM will not support MySQL 8 or later in the future (due to compatibility issues), we recommend to switch to a current version of MariaDB as soon as possible, because Oracle's extended support of MySQL 5.7 will end in 2023 and OpenEMM 22.10 is the last version of OpenEMM officially supporting MySQL 5.7. Please note also, that end of life for MariaDB 10.3 is May 2023.
- **Postfix:** If you use Postfix as MTA and send out high volumes of emails, we recommend to use version 3.4 or later, because it offers a significantly increased sending performance (up to 100%). We have dropped support of Sendmail because while Postfix is actively developed further, progress of Sendmail stalls. Also, beginning with version 3.4, performance of Postfix is superior to Sendmail. See section *Switching SMTP Server from Sendmail to Postfix* in case you still use Sendmail.

3.2 RHEL Operating System

Make sure that service *cron* is enabled on all servers and that ***SELinux* is disabled** by changing parameter *SELINUX* to value "permissive" or "disabled" in file *config* of directory */etc/selinux* and rebooting the server.

Update the operating system to its latest release. This will keep your system in the most stable state and harden it against various intrusion attempts.

Install all required packages. Further dependencies will be resolved and installed automatically by the repository management software.

Install the required packages:

```
# yum update
# yum install gcc make
# yum install xorg-x11-fonts-75dpi fontconfig freetype libX11 libXext
libXrender urw-fonts
```

For the backend of OpenEMM you have to compile and install Python 3.8 or later. But please follow the description in this manual and install the database driver module for Python only after the DBMS is installed. See section *Python 3.8 (or later) for OpenEMM* for details on compilation and deployment of Python 3.

The frontend of OpenEMM needs Java 11. You should use *OMT* to install it, in case it is missing on your server. See section *Runtime Deployment* for details.

If Java 8 is installed on your server, you should remove it with

```
# yum remove java-1.8.0-openjdk
```


© 2023 AGNITAS AG

3.3 SLES Operating System

Make sure that service *cron* is enabled on all servers and that *SELinux* is disabled on all servers.

Update the operating system to its latest release. This will keep your system in the most stable state and harden it against various intrusion attempts.

Install all required packages. Further dependencies will be resolved and installed automatically by the repository management software.

Install the required packages:

```
# zypper install gcc
# zypper install zlib fontconfig libfreetype6 libX11-6 libXext6
libXrender1 xorg-x11-fonts
# zypper install zip sudo wget
```

Get a list of all available repositories:

```
# SUSEConnect -list-extensions
```

For the backend of OpenEMM you have to compile and install Python 3.8 or later. But please follow the description in this manual and install the database driver module for Python only after the DBMS is installed. See section *Python 3.8 (or later) for OpenEMM* for details on compilation and deployment of Python 3.

The frontend of OpenEMM needs Java 11. You should use OMT to install it, in case it is missing on your server. See section *Runtime Deployment* for details.

Install and enable the required logging service:

```
# zypper install rsyslog
# systemctl start rsyslog
# systemctl enable rsyslog
```

Furthermore, directory */usr/sbin* has to be included in the PATH variable for all new users. Therefore, add the following line to file *.profile* in directory */etc/skel* before you create the user for OpenEMM:

```
export PATH=$PATH:/usr/sbin
```


© 2023 AGNITAS AG

4 Python 3.8 (or later) for OpenEMM

Do not install the Python package provided by your operating system. Since the version of Python 3 provided by RHEL or SLES is too old for OpenEMM, you have to compile, build and install version 3.8 (or later) by yourself. **If a former version of Python 3 is already installed on your server, you must de-install it first!**

RHEL option 1:

If you use RHEL 8, there is a shortcut, because you can use a special repository for Python 3.8 so that there is no need to compile and build is yourself.

First, install some required development packages:

```
# yum install gcc gcc-c++ libgcrypt-devel libxml2-devel openssl-devel
```

Activate the repo with

```
# yum module enable python38
```

and install Python 3.8, PIP and all requires packages:

```
# yum install python38 python38-devel python38-pip python38-requests
```

Finally, execute

```
# alternatives --set python3 /usr/bin/python3.8
```

so that "python3" means Python 3.8. Continue with section *Deployment of additional Python 3 modules*.

RHEL option 2:

If you do not want to use the before-mentioned repo for Python 3.8, install the following required packages:

```
# yum install wget gcc gcc-c++ bzip2-devel
# yum install gdbm-devel libgcrypt-devel libffi-devel libxml2-devel
ncurses-devel
# yum install openssl-devel readline-devel sqlite-devel zlib-devel xz xz-
devel
```

SLES:

At first, install the following required packages:

```
# zypper install -y gawk gcc gcc-c++ gdbm-devel libbz2-devel libdb-4_8-
devel libffi-devel libxml2-devel libnsl-devel libopenssl-devel libuuid-
devel make ncurses-devel readline-devel sqlite3-devel tar wget xz-devel
zlib-devel
```

RHEL option 2 + SLES:

Download the compressed tarball of Python 3.8.x (or later) from the official Python website python.org to a working directory of your choice like */root/python3*.

To create and deploy your customized version of Python, execute the following commands as user *root* (with Python 3.8.2 as example):

© 2023 AGNITAS AG

```
# mkdir /home/openemm/opt/Python-3.8.2
# rm -f /home/openemm/opt/python3
# ln -s Python-3.8.2 /home/openemm/opt/python3
# tar -xaf Python-3.8.2.tar.xz
# cd Python-3.8.2
# ./configure --prefix=/home/openemm/opt/Python-3.8.2
# make
# make test
# make install
# export PATH="/home/openemm/opt/python3/bin:$PATH"
# which python3
```

The last command should result in output

```
/home/openemm/opt/python3/bin/python3
```

4.1 Deployment of additional Python 3 modules

Install or update Python's package manager *PIP*:

```
# python3 -m pip install --upgrade pip
```

Finally, install some Python modules required by EMM Inhouse:

```
# python3 -m pip install py3dns
# python3 -m pip install xlrd xlwt xlutils
# python3 -m pip install paramiko pyspf dnspython dkimpy
# python3 -m pip install pycrypto
# python3 -m pip install requests
# python3 -m pip install httpie
# python3 -m pip install setproctitle
# python3 -m pip install inotify
# python3 -m pip install aiodns aiohttp aiohttp-xmlrpc aiosmtpd
# python3 -m pip install msgpack websockets
```

Note: If you have installed MariaDB not from archive.mariadb.org as described in chapter *MariaDB* but from a repository, you have to install the libs that provide the API for third party software, so that the Python module *mariadb* is able to accesss MariaDB.

For **RHEL** execute

```
# yum install mariadb-connector-c mariadb-connector-c-devel
```

(if you get error message that package MariaDB-shared obsoletes these packages, everything is fine, because you have already installed MariaDB-shared before)

and for **SLES** execute

© 2023 AGNITAS AG

```
# zypper install libmariadb-devel
```

5 DBMS for OpenEMM

5.1 MariaDB

If you want to use OpenEMM with a MariaDB database, you have to install the database software before you install the MariaDB database driver module for Python. If MySQL is preinstalled on any of your servers, you have to remove it first before installing MariaDB:

```
# systemctl stop mysql
# yum remove mysql*
```

Since published versions of MariaDB may contain serious bugs including severe security issues (see public bugtracker at https://jira.mariadb.org), we can not recommend to blindly install the latest version available. At press time we use MariaDB **10.6.8** for the EMM public cloud and for us this version works fine. The only significant bug we know is, that you have to add line

```
innodb_stats_persistent = 0
```

in section *[mysqld]* of configuration file *my.cnf* to avoid a known performance bug in this version (see https://jira.mariadb.org/browse/MDEV-28327 for details).

We recommend to install MariaDB from the official archive website https://archive.mariadb.org/. If you want to use the same version we use, download the version mentioned below.

For OpenEMM you need the server and the client component of MariaDB. First create the required group and user:

```
# groupadd mysql
# useradd -g mysql mysql
```

Install required packages which may be missing with

```
# yum install libaio ncurses-compat-libs
```

Create a master configuration file *my.cnf* in directory */etc*:

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
symbolic-links=0
log-error=/var/log/mariadb/mariadb.log
!includedir /etc/my.cnf.d
```

and add a file *client.cnf* in directory */etc/my.cnf.d*:

```
[client]
socket=/var/lib/mysql/mysql.sock
```


© 2023 AGNITAS AG

Create a file path required by *my.cnf*:

```
# mkdir -p /var/log/mariadb/
# chown mysql:mysql /var/log/mariadb/ -R
```

Download the MariaDB tarball, unpack it, create a symlink and change group and owner:

```
# cd /usr/local
# wget https://archive.mariadb.org/mariadb-10.5.8/bintar-linux-systemd-x86_64/mariadb-10.5.8-linux-systemd-x86_64.tar.gz
# tar -zxvf mariadb-10.5.8-linux-systemd-x86_64.tar.gz
# ln -s mariadb-10.5.8-linux-systemd-x86_64 mysql
# cd mysql
# chown -R mysql:mysql .
```

Create the initial MariaDB instance with user *mysql*:

```
# ./scripts/mysql_install_db --user=mysql
```

Execute some more commands for the final configuration:

```
# chown -R root .
# cp /usr/local/mysql/support-files/systemd/mariadb.service /usr/lib/systemd/system/mariadb.service
# chown mysql:mysql /var/lib/mysql
# chmod 755 /var/lib/mysql
```

and open file *mariadb.service* in directory */usr/lib/systemd/system* to change value *LimitNOFILE* to

```
LimitNOFILE=32768
```

To start MariaDB and to make sure it launches every time the server is rebooted:

```
# systemctl enable mariadb
# systemctl start mariadb
```

To be able to use the MariaDB client *mysql*, create shell script *mysql.sh* in directory */etc/profile.d* and insert this line:

```
export PATH=$PATH:/usr/local/mysql/bin/
```

(Log out and log in to your server again to reload the changed environment.)

Last not least, to enable a third party software like Python (used by all backend services) to access the libs of the MariaDB API, create these two symlinks:

```
# ln -s /usr/local/mysql/lib/libmariadb.so.3 /usr/lib64/libmariadb.so.3
# ln -s /var/lib/mysql/mysql.sock /tmp/mysql.sock
```

and add the MariaDB database driver module for Python which allows Python to access the MariaDB API after you have installed Python 3.8 or later. If you have installed Python from a repo as described in option 1 of chapter *Python 3.8 (or later) for OpenEMM*:

```
# python3 -m pip install mariadb
```


© 2023 AGNITAS AG

If you receive an error message with content like *"MariaDB Connector/Python requires MariaDB Connector/C"*, try to install an older version of the module:

```
# python3 -m pip install mariadb==1.0.11
```

If you installed Python manually as described in option 2 of chapter *Python 3.8 (or later) for OpenEMM*:

```
# /home/openemm/opt/python3/bin/python3 -m pip install mariadb
```

If you receive an error message with content like *"MariaDB Connector/Python requires MariaDB Connector/C"*, try to install an older version of the module:

```
# /home/openemm/opt/python3/bin/python3 -m pip install mariadb==1.0.11
```

It is not necessary to create a separate database user for OpenEMM, because the OMT (OpenEMM Maintenance Tool) will take care of it (see section *Runtime Deployment* below).

5.2 MySQL 5.7 for RHEL

We recommend to use MariaDB instead of MySQL, because OpenEMM will support MySQL 5.7 only until version 22.10 and MySQL 8 is not supported! Nevertheless, if you want to use OpenEMM with a MySQL database, you have to install the database software including the shared libraries and development libraries before you install the MySQL database driver module for Python.

After installation of MySQL 5.7 you must finalize the configuration with installing the MySQL database driver module for Python to enable the Python scripts of OpenEMM to access the MySQL database

```
# python3 -m pip install mysqlclient
```

Finally, set a symlink to make sure that OpenEMM can find MySQL:

```
# cd /bin
# ln -s <mysql_home_diectory>/root/bin/* .
```

(Please replace placeholder *<mysql_home_directory>* with the appropriate directory path.)

It is not necessary to create a database user for OpenEMM, because the OMT (OpenEMM Maintenance Tool) will take care of it (see section *Runtime Deployment* below).

5.3 MySQL for SLES

Go to section *MariaDB* before and install MariaDB instead of MySQL.

© 2023 AGNITAS AG

5.4 MariaDB/MySQL Configuration

Set the MariaDB/MySQL root password right after installation with

```
# mysqladmin -u root password '<password>'
```

and save it to file *.mysqlpw* in directory */root* with read and write permission only for user *root*:

```
# cd /root
# vi .mysqlpw
# chmod 600 .mysqlpw
```

MySQL and MariaDB offer several modes of operation. OpenEMM requires the following setting in section *[mysqld]* of DBMS configuration file *my.cnf* (usually located in directory */etc*):

```
sql-mode = "STRICT_ALL_TABLES,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION"
```

Please make sure these three modes (and only these three modes) are set! If the *strict* mode is not selected, if OpenEMM tries to import a string into a database field which is too short to hold the complete string, MariaDB/MySQL would simply cut off the string at the end to make it fit - without any warning! This would harm the integrity of your data.

Preventing *auto-creation* of new database users is a security setting. The ban of *substitutions* makes sure that MariaDB/MySQL uses OpenEMM's choice of database engine InnoDB in any case.

If you want to set property *autocommit* in *my.cnf*, then set it to active:

```
autocommit=1
```

Actually, this is not necessary, because MariaDB/MySQL enables *autocommit* by default.

You might want to change further parameters in configuration file *my.cnf* like commenting out *skip-networking* in section *[mysqld]* or setting *bind-address* to the IP address of your database server.

Especially important are parameter *lower_case_table_names*, which should be set to 1 on Linux systems (to enforce lower case table names), and parameter *wait_timeout*, which is set to 28800 by default. This means that MariaDB/MySQL automatically cuts the connection to OpenEMM after 8 hours of inactivity. This leads to an initial connection error when OpenEMM attempts to contact the MariaDB/MySQL database next time. If your OpenEMM installation does not access its MariaDB/MySQL database all the time, you should increase this value to at least one day (86400) or even to a whole week (604800).

Likewise, you should set *innodb_lock_wait_timeout* to a value of 86400 (default is only 50). This prevents MariaDB/MySQL from canceling a database operation after a mere 50 seconds, which could lead to inconsistent data in the OpenEMM database.

Please do not forget to restart MariaDB/MySQL after changing the configuration file. For more advice on how to configure the database, please check out the database documentation.

© 2023 AGNITAS AG

5.5 DNS Requirements

When setting up the DNS entries for your OpenEMM server, please make sure that your server holds a valid A record and a PTR record which points back to the hostname of your server (see /etc/hosts) for reverse lookups. This is important because most external mailservers that receive emails from your OpenEMM installation will do a reverse DNS lookup in order to check if the FQDN of your server and the PTR record of your server's IP address match. If not, this is an indication of a spambot network and quite often your emails will be rejected.

To be reachable from outside via its FQDN is also important for the GUI service, because this service needs to be able to access itsself via the Internet in order to generate and display preview thumbnails, heatmaps and the like.

If you plan to use an SPF entry for the domain which is used for the sending address of your mass mails, make sure to add your server to this SPF record.

© 2023 AGNITAS AG

6 Server Preparations

Before you are able to install OpenEMM on the server you have to prepare the server first.

Create a group and a user openemm:

```
# groupadd openemm
# useradd -m -g openemm -d /home/openemm -s /bin/bash openemm
# passwd openemm
# su - openemm
```

6.1 Firewall Configuration

Open port 25 and port 8080 in your firewall and add a port forwarding from port 80 to 8080, so you do not have to enter the URL of your OpenEMM server with ":8080" at the end:

```
# firewall-cmd --get-active-zones
```

If your zone is "public" (if not, use the zone name you got with the aforementioned statement):

```
# firewall-cmd --zone=public --add-port=8080/tcp --permanent
# firewall-cmd --zone=public --add-port=25/tcp --permanent
# firewall-cmd --zone=public --add-forward-
port=port=80:proto=tcp:toport=8080 --permanent
# firewall-cmd --reload
```

If you use SUSE, we recommend to use *iptables* which can be installed with

```
# systemctl mask SuSEfirewall2
# systemctl stop SuSEfirewall2
# zypper install iptables
```

6.2 Postfix Deployment

Since you want to use Postfix as SMTP server (MTA), you have to stop and remove Sendmail first (in case it is installed), and you have to install the required packages for Postfix (in case it is not already installed).

RHEL:

```
# systemctl stop sendmail
# yum remove sendmail
# yum install postfix sendmail-milter procmail
```


© 2023 AGNITAS AG

SLES:

```
# systemctl stop sendmail
# zypper remove sendmail
# zypper install postfix procmail libmilter1_0
```

RHEL + SLES:

Further dependencies will be resolved and installed automatically by the repository management software.

Switch the default SMTP server to Postfix with

```
# alternatives --set mta /usr/sbin/sendmail.postfix
```

and create a symlink so that OpenEMM can find the Postfix mail log file:

```
# ln -s /var/log/mail /var/log/maillog
```

After installation of Postfix, you have to change its configuration to unleash all features. To do this, change to the Postfix main configuration directory:

```
# cd /etc/postfix
```

Add some configuration parameters to Postfix' main configuration file *main.cf*:

```
inet_interfaces = all
inet_protocols = all
mailbox_command = /usr/bin/procmail
mailbox_size_limit = 0
message_size_limit = 0
maximal_queue_lifetime = 1d
bounce_queue_lifetime = 1d
smtp_tls_security_level = may
smtp_tls_protocols = !SSLv2, !SSLv3
smtp_tls_ciphers = high
smtp_tls_mandatory_ciphers = $smtp_tls_ciphers
hash_queue_depth = 2
enable_long_queue_ids = yes
relay_domains = /home/openemm/var/run/relay.domains
transport_maps = hash:/home/openemm/var/run/transport.maps
smtpd_milters = unix:/home/openemm/var/run/bav.sock
```

If lines with parameters of the same name already exist in file *main.cf* (like *inet_interfaces, inet_protocols* or *smtpd_tls_security_level*), comment them with character # at the beginning to avoid any warning messages or overwrite them with the new values in case you do not want to keep the original values as backup.

The two files *relay.domains* and *transport.maps* as well as Linux socket *bav.sock* are automatically created at first startup time of the mailloop service. File *relay.domains* specifies your mailloop service domain name, so that responses like auto-replies and bounces sent to an email address with this domain name are accepted by Postfix for relaying. File *transport.maps* defines for the mailloop service domain name the service used for processing.

© 2023 AGNITAS AG

Additionally, you have to set parameter *myhostname* in file *main.cf* to the FQDN of your OpenEMM server like *openemm.domain.com*. Otherwise, mails would be sent with sender domain *localhost.loacldomain* instead.

If you want to be able to receive autoresponder, bounce and feedback mails encrypted with the TLS protocol, add

```
smtpd_use_tls = yes
smtpd_tls_loglevel = 2
smtpd_tls_security_level = may
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
smtpd_tls_eecdh_grade = strong
smtpd_tls_cert_file = <path_to_CERT_file>
smtpd_tls_key_file = <path_to_KEY_file>
smtpd_tls_CAfile = <path_to_CERT_chain>
smtpd_tls_CApath = <path_to_CERT_directory>
smtpd_tls_protocols = !SSLv2, !SSLv3
smtpd_tls_ciphers = high
```

to file *main.cf*. Take care to replace the four placeholders with the real directory paths to the specified files to make sure that Postfix is able to receive TLS encrypted mails. Certificate *mail.crt* may be a self-signed certificate.

Finally, the configuration parameters for service "mailloop" are defined in configuration file *master.cf*. Add these two lines:

```
mailloop unix - n n - - pipe
  flags=RX user=openemm argv=/usr/bin/procmail /home/openemm/lib/bav.rc
```

Please do **not** omit the two space characters before keyword "flags" to indicate the parser that the line is continued!

Last not least, activate the TLS manager in file *master.cf* by uncommenting (i.e. removing the leading #) line

```
tlsmgr unix - - n 1000? 1 tlsmgr
```

To activate all changes, restart Postfix with

```
# systemctl restart postfix
```

In case you start Postfix separately from OpenEMM and Postfix complains that file *relay.domains* is missing, you can ignore this warning because OpenEMM take care to create this file at startup time in case it is missing.

6.3 Tomcat Deployment

OpenEMM needs web application engine Tomcat for the frontend services (GUI, statistics and webservices). Tomcat can be installed with OpenEMM Maintenance Tool *OMT*. For details please read section *Runtime Deployment* below.

© 2023 AGNITAS AG

If you want to operate OpenEMM with the HTTPS protocol, the server key files (**.key*, **.pem*) and server certificate files (**.crt*, *cacerts*) for the TLS configuration (to allow HTTPS connections) must be provided from your side as these files are server and client specific. See section *Apache Native Library* for details.

6.4 Image and PDF Creation Tool wkhtmltox

In order to create good looking thumbnail images and PDF documents in the OpenEMM GUI, the OpenEMM server needs the powerful tool *wkhtmltox* to be installed. For RHEL, the tool can be installed with OpenEMM Maintenance Tool *OMT* (see section *Runtime Deployment* below), or you can do it manually:

RHEL:

```
# yum install zlib fontconfig freetype libX11 libXext libXrender
```

RHEL 7:

```
#yum install
https://github.com/wkhtmltopdf/packaging/releases/download/0.12.6-1/
wkhtmltox-0.12.6-1.centos7.x86_64.rpm
```

RHEL 8:

```
# yum install
https://github.com/wkhtmltopdf/packaging/releases/download/0.12.6-1/
wkhtmltox-0.12.6-1.centos8.x86_64.rpm
```

If a later version of wkhtmltox is available, you may use the most recent one. But do not use the *wkhtmltox* package of the operating system, because it may be buggy.

SLES:

At press time, no current build of *wkhtmltox* was available for SLES. Therefore, you have to use version 0.12.4, which still has a generic build. You can get *wkhtmltox* at GitHub, and you can install it in */opt* and create a symlink in */usr/local/bin*:

```
# cd /opt
# wget
https://github.com/wkhtmltopdf/wkhtmltopdf/releases/download/0.12.4/
wkhtmltox-0.12.4_linux-generic-amd64.tar.xz
# tar xJpf wkhtmltox-0.12.4_linux-generic-amd64.tar.xz
# rm wkhtmltox-0.12.4_linux-generic-amd64.tar.xz
# chown -R root. wkhtmltox
# ln -s /opt/wkhtmltox/bin/wkhtmlto* /usr/local/bin/
```

To be able to create thumbnails and PDF files, the OpenEMM frontend needs to access itsself to execute *wkhtmltox*. If the OpenEMM server can not access itsself via port 80, you have to define the working URL for frontend access (like https://openemm.domain.com:8443) in the database with

© 2023 AGNITAS AG

```
INSERT INTO config_tbl (class, name, value) VALUES ('preview', 'url',
'<URL for frontend access including required port>');
```

6.5 Configuration of Operating System Logging Parameters

With version 7 of RHEL and version 12 of SLES a new "feature" was introduced that drops messages from being logged if the server has a high workload. If you use the default configuration values of RHEL or SLES (i.e. 1,000 entries max. within 30 seconds), and if the OpenEMM server has a high workload due to a high mail output, this can lead to missing entries in the maillog. However, missing entries in the maillog mean that OpenEMM does not know whether mails were delivered successfully or not and may lead to an incomplete bounce management and incomplete statistics!

To make sure that even under high workload all messages are logged to the maillog, we recommend to change (or add) the following values of file *journald.conf* in directory */etc/systemd*:

```
RateLimitIntervalSec=10s
RateLimitBurst=20000
```

(RHEL 7: Use *RatelimitInterval* instead of *RateLimitIntervalSec*.)

Afterwards, restart the journal daemon with

```
systemctl restart systemd-journald
```

to activate your changes.

RHEL 7 and SLES:

Additionally, add (or change) the following values in file *rsyslog.conf* of directory */etc* after the line with parameter *$IMJournalStateFile imjournal.state*:

```
$imjournalRatelimitInterval 10
$imjournalRatelimitBurst 20000
```

RHEL 8:

Change in file *rsyslog.conf* in directory */etc* lines

```
module(load="imjournal"             # provides access to the systemd journal
    StateFile="imjournal.state")    # File to store the position in the journal
```

to the following one-liner:

```
module(load="imjournal" StateFile="imjournal.state" ratelimit.interval="10"
ratelimit.burst="20000")
```

RHEL+ SLES:

Afterwards, restart the rsyslog service to activate your changes in file *rsyslog.conf*:

```
systemctl restart rsyslog
```


© 2023 AGNITAS AG

6.6 Miscellaneous

OpenEMM needs a minimum value of 16384 for kernel parameter *nofile*, which defines the maximum number of open files per process. OpenEMM Maintenance Tool *OMT* will check and change if necessary.

OpenEMM requires read access to the mail log file at */var/log/maillog* and *logrotate* has to be aware of this fact, too. Open file *syslog* in directory */etc/logrotate.d* and add the following line after the line *sharedscripts*:

```
# create 0644
```

and run

```
# chmod 644 /var/log/maillog
```

to set the permissions of the current maillog.

© 2023 AGNITAS AG

7 OpenEMM Deployment

7.1 Download

We recommend to download the deployment & runtime package of OpenEMM from https://www.agnitas.de/en/download/openemm-binaries/. This package provides an installer tool for the backend and frontend code and eases installation, updates, administration and maintenance of OpenEMM significantly.

Of course, you can also download the source code of OpenEMM from GitHub at https://github.com/agnitas-org/openemm and compile and deploy the software manually yourself. See the build instructions at the end of the OpenEMM wiki page at https://wiki.openemm.org.

7.2 Runtime Deployment

Download the OpenEMM runtime tarball (file name: *openemm-runtime-20.10.*.tar.gz*) to */home/openemm* to create the deployment and runtime environment for OpenEMM. Change to user openemm, unpack the tarball and start it with

```
# su - openemm
$ tar -xvzpf openemm-runtime-20.10.*.tar.gz
$ OMT.sh
```

At the first start OMT checks for certain packages and settings and offers to change them.

While OpenEMM needs Python 3.8 or higher to work, *OMT* itsself is fine with any version 3 of, i.e. the python3 package of the operating system would be sufficient.

OMT is an interactive command line tool, which should be your preferred way to check, configure and update various components of OpenEMM.

```
===============================================
= OpenEMM Maintenance Tool (OMT) v21.10.009 =
===============================================
Root mode: On
Hostname: openemm-next.agnitas.de
OpenEMM License: OpenEMM (ID: 0)
OpenEMM Runtime Version: 21.10.009
OpenEMM Version: 21.10.000.034
System-Url: https://oe.agnitas.de

Current menu: Main

Please choose (Blank => Quit):
```


© 2023 AGNITAS AG

```
 1. Show OpenEMM status
 2. Configuration
 3. Database Maintenance
 4. Security
 5. Install or update package from AGNITAS Website
 6. Install or update package from local file
 7. Install or update package from AGNITAS Cloud
 8. Show update history
 9. Switch OpenEMM version
10. Restart OpenEMM
11. Send configuration and log files in email
 0. Quit
>
```

At launch time *OMT* checks if kernel parameters in */etc/security/limits.conf* are big enough and, if not, offers to change them. To be able to change this file, you need to start *OMT* as user *root*. After the change of the kernel parameters, you have to reboot the server.

At launch time *OMT* also checks your software stack and offers to install required packages (including Java 11, including Tomcat 10 and Tomcat-Native for HTTPS support) and to set environment variables

- JAVA_HOME (default path: */usr/lib/jvm/java*) for Java
- CATALINA_HOME (default path: */home/openemm/opt/tomcat/* for Tomcat
- Tomcat-Native (dfault path: */home/openemm/opt/tomcat-native*)
- WKHTMLTOPDF (default path: */usr/bin/wkhtmltopdf*) for PDF tool wkhtmltox
- WKHTMLTOIMAGE (default path: */usr/bin/wkhtmltoimage*) for image tool wkhtmltox

These parameters will be written to file *setenv.sh* in directory */home/openemm/bin/*.

If one of these tools is not available on your server yet, *OMT* offers to install it for you. Note: If you want to install Java or tool wkhtmltox, you have to launch *OMT* as user *root*:

```
# sudo su -
# /home/openemm/bin/OMT.sh
```

OMT helps you to set up and configure a database connection (file *dbcfg*) including database user and password. In menu *Configuration → Change database configuration* you can enter the required properties like database type ("mysql" or "mariadb"), database name ("openemm"), database user (suggestion: "openemm") and your database password. If the database configuration file *dbcfg* does not exist in directory */home/openemm/etc/*, it is also created from scratch. A valid *dbcfg* file looks like this:

```
openemm: dbms=mariadb, name=openemm,
jdbc-connect=jdbc:mariadb://127.0.0.1/openemm?
zeroDateTimeBehavior=convertToNull&useUnicode=true&characterEncoding=UTF-
8, host=127.0.0.1, user=openemm, jdbc-driver=org.mariadb.jdbc.Driver,
password=<password>
```


© 2023 AGNITAS AG

If the OpenEMM database and its user does not exist, *OMT* offers to set it up for you after you have entered the properties mentioned above. The empty OpenEMM database will be filled later during installation of the OpenEMM code package.

If for whatever reason the creation of the database user for OpenEMM does not work, you can create the user manually with

```
# mysql -u root -p
MariaDB > CREATE USER 'openemm'@'localhost' identified by '<password>';
MariaDB > GRANT ALL PRIVILEGES ON `openemm`.* TO 'openemm'@'localhost' ;
MariaDB > GRANT SELECT ON mysql.proc TO 'openemm'@'localhost';
MariaDB > GRANT RELOAD ON *.* TO 'openemm'@'localhost';
MariaDB > GRANT SUPER ON *.* TO 'openemm'@'localhost';
```

7.2.1 Menus of OMT

Menu "Show EMM status" starts a brief health check of the OpenEMM configuration on the current server.

Menus "Configuration" help you to check, change or create certain settings of your OpenEMM configuration without having to deal directly with the OpenEMM database or configuration files:

```
Current menu: Configuration

Please choose (Blank => Back to Main):
  1. Configure basic environment (Java, Tomcat, Wkhtml, Proxy)
  2. Change configuration of database connection
  3. Change basic configuration
  4. Change client/account configuration
  0. Back to Main
 >
```

With menu "Security" you can set a new password for admin user "emm-master" if you select option "Create new initial 'emm-master' password". You may use this option also in case you have forgotten the password for user *emm-master*.

Menu Database Maintenance offers a backup and restore of the database present on the server.

Menu "Install or update package from ..." supports you in downloading packages of a new OpenEMM release from various sources. Please be aware that you have to start the *OMT* as user *root* to be able to install or update a backend service. If you want to load a file from an external source (AGNITAS), make sure that the server *OMT* is running on, is able to access this source. Otherwise download the file first with a different device, copy it to your server and install it as local file.

© 2023 AGNITAS AG

If you need to know which services and versions of OpenEMM were active at what time in the past, menu Show update history provides a list of all available services, their versions and the exact startup times.

Menu "Switch OpenEMM version" lets you roll back to a former version of an OpenEMM service or roll forward to a later version. However, if you switch back not to an older patch level (last 3 digits of the version number), but to an older minor or major version (like going back from 20.10 to 20.04), please be advised that the database schema is not rolled back and, therefore, some feature may not work properly because it wants to use a database field that does not exist any longer in the new database schema.

After a change of version or after an OpenEMM service update, use menu "Restart OpenEMM" to restart the OpenEMM installation on the current server (frontend and backend services, if available).

Menu "Send configuration and log files in email" helps you to create an email which sends all important OpenEMM configuration information to an email address of your choice for diagnostic purposes or as a backup. For that reason, the tool collects certain configuration and log files from the current server as well as the content of database configuration tables of your OpenEMM installation and sends them out as zipped and password-protected attachment. Since this attachment can become quite big, make sure that the receiving email address is able to handle this payload.

7.3 OpenEMM Deployment

For the deployment of the openemm code tarball you have to start *OMT* as user *root*, because some files of the tarball have to be deployed with root permissions:

```
$ sudo su -
# /home/openemm/bin/OMT.sh
```

Use menu *Install or update package from AGNITAS website* to install the openemm code package. As first step, the menu offers to install an update of the runtime package, if a later version is available. Opposed to upgrading to a new major version of OpenEMM, an update to a new runtime version is always recommended.

After that, versions of Tomcat and the Tomcat Native library are offered, but you can always select "n" for any package, if you do not want to install it or if it already exists. Please be aware that even if you install the Tomcat native library, you have to take care of its configuration to be able to use the HTTPS protocol for OpenEMM (see section *Apache Native Library* for details).

Answer “no” to all downloads offers until the openemm code package is offered (if you started *OMT* as user *root*). If an upgrade of a major version of OpenEMM is available (like 21.04) this upgrade opportunity is offered first. However, you should not blindly accept it if you are working on a production system (see section *Test and Production System* below).

© 2023 AGNITAS AG

During Deployment of the openemm code tarball several new directories and symlinks will be created in directory */home/openemm/*. Also, the OpenEMM database is automatically populated with its initial content.

If you do not want to install or update the code of OpenEMM, you can start *OMT* as user *openemm*:

```
# su - openemm
$ OMT.sh
```

7.4 Configuration

After deployment of the OpenEMM code but before launching it with the restart menu, use menu *Configuration → Change basic configuration* to set up and change the configuration of OpenEMM. Note: It is possible to launch OpenEMM without adapting the configuration, but in this case not all features of the software will work properly.

With OpenEMM 20.04 all configuration properties have been migrated from files *emm.properties, emm-ws.properties* and *Mailgun.ini* to the OpenEMM database. Therefore, you do not change these files any longer but the corresponding database entries listed in the sub-menu mentioned above. The advantage of the migration from files to database is, that further updates of OpenEMM do not overwrite any individual settings you might have made.

Sub-menu *Change basic configuration* asks you to enter the full URL of your server including protocol (http or https). From this value it deviates the settings for various essential properties.

Sub-menu Change client/account configuration offers you to change the redirect domain and the mailloop domain. These value are pre-filled by the basic configuration and you should change those values only if you know what you are doing:

- Set *rdir_domain* to the protocol and FQDN of your server, for example *https://openemm.example.com*. This domain will be used in all measureable links to redirect them through OpenEMM. This property **must** include the appropriate protocol.
- Set *mailloop_domain* to the domain of your sender address. The domain for the mailloop service must be different from the name of your OpenEMM server. It usually is the FQDN which is defined in the MX record for your server, for example *mailing.example.com*, pointing as MX to *openemm.example.com*. In this case, use domain *mailing.example.com* as domain for the sender address of your mailings. (There are other ways for configuration, described in section *Mailloop Configuration*.) If you do not configure the mailloop service, OpenEMM can only process instant bounces, i.e. you will not be able to catch all bounces. This property **must not** include any protocol.

Finally, create an initial password for admin user "emm-master" in menu *Security → Create new initial 'emm-master' password*. Write down the generated password and use user "emm-master" and this initial password later for your first login in the GUI of OpenEMM. At first login, you will be prompted to change your password to a new one of

© 2023 AGNITAS AG

your choice. In case you forget your password for the admin user at a later time, you can always use this menu of *OMT* to set a new password.

7.5 Startup

Launch OpenEMM with menu *Restart OpenEMM* of *OMT*. Experts may have a look at the Tomcat log during startup with

```
$ tail -f /home/openemm/logs/catalina.out
```

to check for any warnings or error messages. Be patient, because the startup process will take a minute or two.

After OpenEMM has been lauched successfully, point your browser to your OpenEMM server and log in with user “emm-master” and the initial password you just created. OpenEMM will ask you to change the initial password to a new one of your choice.

If you want to install the context-sensitive online help feature of OpenEMM (including the user manual with about 500 pages), visit

- https://www.agnitas.de/en/download/openemm-manual/ (English)
- https://www.agnitas.de/download/openemm-handbuch/ (German)

to get the download link for the manual package. You can use this link in *OMT's* menu *Install or update package from AGNITAS Cloud* to install the documentation.

7.6 Test and Production System

If OpenEMM is an important software for you (and we hope it is!) we recommend to set up a separate test system alongside your production system. This should be no probem for you, because the complete software stack for OpenEMM is open source and you do not have to pay a dime for the additional instance.

You can create a copy of your OpenEMM production database with menu *Database Maintenance → Backup MariaDB/MySQL database* of OpenEMM Installer. On your test system you should only create the empty OpenEMM database with menu *Configuration → Change configuration of database connection*. Now you can set up the database copy of your production system with menu *Database Maintenance → Restore MariaDB/MySQL database*. But make sure that both systems use the same DBMS version to avoid any trouble.

After that, you only have to adjust the properties containing the server URL in menu *Configuration → Change basic configuration*, because they have been copied from your production database as well. Now you are ready to use your test system to try out different configuration settings, to test database backup and restore or to test an upgrade to a new major OpenEMM version – all that before applying any change to your precious production system.

© 2023 AGNITAS AG

8 OpenEMM Testing

For testing (and to initialize the system) you should first make sure that MariaDB (or MySQL) contains the OpenEMM database and the DBMS is already up and running.

If you are not sure about OpenEMM's status use the first menu *Show OpenEMM status* of *OMT* to check it and use menu *Restart OpenEMM* to restart the software if it does not look right.

To test the correct operation of each service we recommend

- creating a new mailing (testing the GUI of the frontend)
- sending the mailing to an existing email address (testing the backend)
- and sending it to an non-existing email address (testing the bounce management)*
- opening the mail and clicking the links (testing the redirect service of the frontend)
- checking in mailing statistics if openings and clicks were recorded (testing the redirect service and statistics service of the frontend)

*To test the bounce management you have to send the mailing not to admin or test recipients, but to normal recipients, because due to performance reasons mailings to admin and test recipients are sent out instantly without the overhead of a regular mailing. For that reason the bounce management process can not be applied to mailings for admin and test recipients.

8.1 OpenEMM Does Not Send Emails

The most "popular" problem of OpenEMM is, that no emails are sent out. Therefore, we have compiled a checklist with the ten most common reasons why OpenEMM does not send emails:

1. Packages *postfix* and *sendmail-milter* are not installed.
2. Postfix has not been started ("systemctl start postfix").
3. Postfix is no longer running ("ps aux | grep postfix/master").
4. The wrong MTA is active (Sendmail, qmail, Exim, etc.).
5. Port 25 has not been opened.
6. Service *cron* is disabled ("systemctl status crond").
7. SELinux is enabled.
8. Postfix' mailqueue at */var/spool/postfix* does not exist (solution: re-install Postfix).
9. For reverse lookups, no PTR record does exist for the IP address of the OpenEMM server (so that mails are blocked as spam by mailbox providers on the receiving end).
10. Postfix' mail log file in */var/log/maillog* shows errors.

© 2023 AGNITAS AG

9 OpenEMM Updates and Upgrades

With OpenEMM we make a difference between **updates** (changing to a new minor release, like from 20.10.000.050 to 20.10.000.100) and **upgrades** (changing to a new major release, like from 20.04 to 20.10).

If you plan to do an **upgrade**, we strongly recommend to **update** to the lastet minor release of your OpenEMM version, first. Only after that, execute the upgrade to the new major version. This makes sure, that the delta between the two major versions is as little as possible and the old version is well prepared for the upgrade.

If you do not operate a test environment for OpenEMM, but you apply updates and upgrades right to the production environment, **make sure to create a backup with OMT → Database Maintenance → Backup MariaDB/MySQL database before** (even more so, if you upgrade to a beta version of OpenEMM), because you might need the database backup in case of a roll back (see section *Rollbacks for OpenEMM* below for details).

9.1 Updating and Upgrading OpenEMM

If you plan to upgrade OpenEMM 21.04, please check sections *Special Advice for Upgrading to 21.04* and *Special Advice for Upgrading to 21.10* below for required updates to your current software stack.

The OpenEMM Wiki at https://wiki.openemm.org shows the latest available versions of the runtime and openemm code package at the top.

You can download the latest versions of the packages with menu *Install or update package from AGNITAS Website* of *OMT*. Please answer “no” to all downloads offers until the openemm code package is offered.

An update of the openemm code package will also update the schema of the OpenEMM database, if necessary. Please keep in mind that you have to start *OMT* as user *root*, if you want to update the openemm code package.

Do not forget to restart OpenEMM with menu *Restart OpenEMM* after you have installed a new openemm code package to activate it. But do the restart at an convenient time: Do not restart OpenEMM during a dispatch of a mailing, or right after the dispatch (due to the brief downtime of the redirect service which would cause missed openings and click redirects).

9.2 Do not Skip Major Releases when Upgrading

When updating OpenEMM, do not skip a major release, i.e. if you use OpenEMM 20.10 and you plan to upgrade to OpenEMM 21.10, **do not skip** the upgrade to version 21.04! You have to insert upgrades to the intermediate major versions in order to make sure that no migration processes for the OpenEMM database are missing.

To make sure that you are able to upgrade to each major release of OpenEMM, update to the runtime version of the next major relase and after that download and update the openemm code package. **Do not** update the runtime package to a new major release **without** updating the openemm code package, too!

© 2023 AGNITAS AG

If you have skipped updating openemm code packages while updating the runtime package, you have to use the switch menu to switch back to an older runtime version (see section *Rollbacks for OpenEMM* below for details).

Note: If you need to switch back to a 20.04 release and you are not offered runtime 20.04.050 or later, please make a copy of your individual file *server.xml* in directory */home/openemm/conf* because runtimes before 20.04.050 used to overwrite the existing *server.xml* when switching back.

Even if you do not want to use an intermediate version, you have to restart this version (like 20.04 in the above-mentioned case) once. This makes sure that the startup process in the OpenEMM code is executed and initiates all pending migration tasks contained in the skipped OpenEMM version. For example, the startup process of OpenEMM 20.04 migrates all configuration properties from configuration file *emm.properties* into the database (among other things). This migration is important, because OpenEMM 20.10 reads its configuration properties from the database.

9.3 Special Advice for Upgrading to 21.04

Starting with release 21.04, OpenEMM uses the MariaDB database driver instead of the MySQL database driver, if a MariaDB is used for OpenEMM. Therefore, before you upgrade to OpenEMM 21.04, you need to install some additional packages and modules.

RHEL 7:

If you used the MariaDB packages of RHEL (rh-mariadb102) before, you have some work to do! First, you need to include the official MariaDB repository. Use link https://downloads.mariadb.org/mariadb/repositories to find the repository suitable for you and follow the instructions by MariaDB.

At first, make a backup of your OpenEMM database – just in case!

After that, you have to uninstall MariaDB and re-install MariaDB from the official MariaDB repository.

Stop OpenEMM and MariaDB:

```
# sudo -u openemm /home/openemm/bin/openemm.sh stop
# systemctl stop rh-mariadb102-mariadb
# systemctl disable rh-mariadb102-mariadb
```

Make sure, that directory */var/lib/mysql/* is empty and move the MariaDB files there:

```
# ll /var/lib/mysql/
# mv /var/opt/rh/rh-mariadb102/lib/mysql /var/lib/
```

Install the official MariaDB repository:

```
# vim /etc/yum.repos.d/MariaDB.repo
```

This is an example for our settings:

```
# MariaDB 10.5 CentOS repository list - created 2021-04-12 12:48 UTC
# http://downloads.mariadb.org/mariadb/repositories/
[mariadb]
```


© 2023 AGNITAS AG

```
name = MariaDB
baseurl = http://yum.mariadb.org/10.5/centos7-amd64
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck=1
```

Now, you can install the new version of MariaDB:

```
# yum install MariaDB-server MariaDB-client MariaDB-shared MariaDB-devel
```

Copy the old configuration file *mc.cnf* in directory */etc/opt/rh/rh-mariadb102/* to directory */etc* and check the settings.

```
# cp /etc/opt/rh/rh-mariadb102/my.cnf /etc/my.cnf
# cat /etc/my.cnf
```

To clean up, remove the old MariaDB and start the new one:

```
# yum remove rh-mariadb*
# systemctl start mariadb
# systemctl enable mariadb
# systemctl status mariadb
```

Finally, execute the database upgrade script with

```
# mysql_upgrade -u root -p
```

to update all tables of the database for the new version of MariaDB.

RHEL 8 (in case you are using MariaDB included in the distribution):

```
# yum install mariadb-connector-c mariadb-connector-c-devel
```

If you get error message that package *MariaDB-shared* obsoletes these packages, everything is fine, because you have already installed *MariaDB-shared* before.

SLES 15 (in case you are using MariaDB included in the distribution):

```
# zypper install libmariadb-devel
```

RHEL+SLES:

You must install the Python 3 module for MariaDB! If you installed Python 3 from a repo as described in section *Python 3.8 (or later) for OpenEMM*:

```
# python3 -m pip install mariadb
```

If you receive an error message with content like *"MariaDB Connector/Python requires MariaDB Connector/C"*, try to install an older version of the module:

```
# python3 -m pip install mariadb==1.0.11
```

If you installed Python manually as described in section *Python 3.8 (or later) for OpenEMM*:

```
# /home/openemm/opt/python3/bin/python3 -m pip install mariadb
```


© 2023 AGNITAS AG

If you receive an error message with content like *"MariaDB Connector/Python requires MariaDB Connector/C"*, try to install an older version of the module:

```
# /opt/agnitas.com/software/python3/bin/python3 -m pip install
mariadb==1.0.11
```

Finally, you have to change the settings in configuration file *dbcfg* from MySQL to MariaDB. Use OMT's menu *Configuration*, sub-menu *Change configuration of database connection* to change parameter *jdbc-connect* to

```
jdbc:mariadb://127.0.0.1/openemm?
zeroDateTimeBehavior=convertToNull&useUnicode=true&characterEncoding=UTF-
8
```

and parameter *jdbc-driver* to

```
org.mariadb.jdbc.Driver
```

9.4 Special Advice for Upgrading to 21.10

Beginning with version 21.10, OpenEMM is compiled with Java 11, requires Java 11 as runtime environment and Tomcat 10 due to its Jarkata EE support. (Tomcat 9 does not work with byte code created by Java 11 and does not support Jarkata EE). You may use *OMT* to install Java 11 and Tomcat 10 or read section *Tomcat Deployment* before.

Also, OpenEMM does not support Sendmail any longer because because while Postfix is actively developed further, progress of Sendmail stalls. Furthermore, Postfix offers easier configuration, better performance and better logging. If you still use Sendmail, we strongly recommend to switch to Postfix instead. See section *Switching SMTP Server from Sendmail to Postfix* below for details.

If you use CentOS 8, you probably have already learned that this version will be discontinued by the end of 2021. Therefore, for servers using CentOS 8, we recommend a migration to AlmaLinux. The migration of a server from CentOS 8 to AlmaLinux 8 is quite easy and straightforward:

At first, make a backup of your server. We could not test all possible scenarios, so there is a risk that something goes wrong. In such a situation you will have a restore point.

Download the AlmaLinux migration script *almalinux-deploy.sh* with

```
curl -O
https://raw.githubusercontent.com/AlmaLinux/almalinux-deploy/master/
almalinux-deploy.sh
```

Execute this script and check its output for errors:

```
# bash almalinux-deploy.sh
 ...
Migration to AlmaLinux is completed
```

Make sure that your system was successfully converted by checking the release file:

© 2023 AGNITAS AG

```
# cat /etc/redhat-release
.
AlmaLinux release 8.4 (Electric Cheetah)
```

And check that your server boots the AlmaLinux kernel by default:

```
# grubby --info DEFAULT | grep Alma
.
title="AlmaLinux (4.18.0-305.el8.x86_64) 8.4"
```

9.5 Special Advice for Upgrading to 22.04

This version does not need any special advise. But if you skipped a version, please see the section before.

9.6 Special Advice for Upgrades to Version 22.10

OpenEMM 22.10 needs two more Python packages on all servers with a backend service. If you installed Python 3.8 from the repo (see section *Python 3.8 (or later) for OpenEMM*), use

```
# pip3.8 install msgpack
# pip3.8 install websockets
```

If you compiled Python yourself, use

```
# python3 -m pip install msgpack
# python3 -m pip install websockets
```

Please note that this is the last version supporting MySQL. If you want to migrate from MySQL to MariaDB, we recommend this procedure:

1. stop all EMM services and MySQL
2. export the database content to have a backup just in case (for instance using command *mysqldump*)
3. change your DBMS from MySQL 5.7 to MariaDB 10.2 (because these versions are compatible: https://mariadb.com/kb/en/upgrading-from-mysql-to-mariadb)
4. run mysql_upgrade
5. change file *dbcfg* in directory */opt/agnitas.com/etc* for MariaDB
6. start MariaDB
7. start EMM services

If the EMM database does not work correctly, import it from your backup instead (for instance using command *mysql*).

© 2023 AGNITAS AG

9.7 Rollbacks for OpenEMM

If you are not happy with a version you updated or upgraded to, you can roll back OpenEMM to a former version or roll forward to a later version with menu *Switch OpenEMM version*. This menu lists the active versions of the EMM services available on the current server so that you can switch separately

- the runtime environment *RUNTIME* (including OMT itsself)
- the openemm frontend (consisting of GUI service *EMM*, statistics service *STATISTICS* and webservices service *WS*)
- the openemm backend (consisting of various services packaged into *BACKEND-OPENEMM*)
- the documentation package *MANUAL* (including all context sensitive help pages)

After you have selected a specific service you get a list of all versions available on the server so that you can chose the version you want to activate.

Warning: If you switch back the openemm frontend or backend not to an older minor version (last 3 digits of the version number), but to an older major version (like going back from 20.10 to 20.04), be advised that the database schema can not be rolled back and, therefore, some feature may not work properly in case it wants to use a database field that does no longer exist in the new database schema.

Therefore, if you want to switch back OpenEMM to an older major version, we recommend to restore the database backup of the corresponding version.

If you need to know which services and versions of OpenEMM were active at what time in the past, menu *Show update history* of *OMT* provides a list of all available services, their versions and the exact startup times.

Do not forget to restart OpenEMM with menu *Restart OpenEMM* after you have switched the frontend or backend version of OpenEMM.

If you get offered updates only to the latest version of OpenEMM, you have to switch back the runtime version. For instance, a runtime version 20.10 only offers you updates to versions of OpenEMM 20.10. But as soon as you have upgraded to a runtime version 21.04, you get offered upgrades to OpenEMM 21.04 as well.

9.8 Templates and Web Forms

You do not have to start from scratch when producing mailings or creating web forms in OpenEMM. At https://www.agnitas.de/en/download-center/ you can download templates and web forms which you can import into OpenEMM. **Make sure to replace in web forms any placeholders for a company ID with value "1" if this is not done automatically during import.**

© 2023 AGNITAS AG

9.9 Preparations before Updating MariaDB

If you plan to update your version of MariaDB, due to a bug in older versions of MariaDB (MDEV-19292), you may get an error "rowsize too large" afterwards which prevents you from using your OpenEMM database after the update of MariaDB.

Statement from MariaDB regarding this problem:

"Prior to MariaDB 10.2.26, MariaDB 10.3.17, and MariaDB 10.4.7, MariaDB doesn't properly calculate the row sizes while executing DDL. In these versions, unsafe tables can be created, even if InnoDB strict mode is enabled."

To check if you are affected by this bug, please execute SQL statement

```
SQL> SELECT count(*) FROM information_schema.innodb_sys_tables WHERE name
LIKE 'openemm/%' AND ROW_FORMAT = 'Compact';
```

If the result of this SQL statement is greater than 0, your OpenEMM database uses tables using the format which may cause "unsafe" tables. **In this case, we strongly recommend to convert all database tables to format "dynamic" with statement**

```
ALTER TABLE <table name> ROW_FORMAT = DYNAMIC;
```

before upgrading your version of MariaDB (after you made a backup of your database, of course)!

To simplify this task for you, *OMT* offers an automated conversion in menu *Database Maintenance*, sub-menu *Check MariaDB table format*. Alternatively, here is a little script that does the conversion of all tables (please execute as database user *root*):

```
DELIMITER $$
create procedure tmp_convert_row ()
Begin
   DECLARE done INT DEFAULT FALSE; DECLARE dbtable varchar(100);
   declare tab_curse cursor for select SUBSTRING(name, 9) from
information_schema.innodb_sys_tables where name like 'openemm/%' and
ROW_FORMAT = 'Compact';
   DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
    open tab_curse;
     read_loop: LOOP
        fetch tab_curse into dbtable;
    IF done THEN LEAVE read_loop; END IF;
    SET @SQLText = CONCAT('alter table ', dbtable, '
ROW_FORMAT=DYNAMIC');
        PREPARE stmt FROM @SQLText; EXECUTE stmt; DEALLOCATE PREPARE
stmt;
   end loop;
   close tab_curse;
END$$
DELIMITER ;
call tmp_convert_row ();
```


© 2023 AGNITAS AG

10 Advanced Configuration

This chapter describes some more advanced configuration options. Please be aware that you should backup first any files you modify in case the configuration is broken afterwards.

10.1 Mailloop Configuration

The mailloop service enables OpenEMM to process bounces (and autoresponder mails) which can be sent back to the sender address hours or even days after OpenEMM dispatched the mailing. We call these late bounces “asynchronous bounces”.

You need to define a dedicated mailloop service domain name which is different from the OpenEMM server hostname (set in */etc/sysconfig/network/*). While you have to set up the A record for the OpenEMM server hostname, you have to set an MX (Mail Exchange) record for the mailloop service domain name, which points to the OpenEMM server hostname for correct mail routing.

For each new bounce filter created in the OpenEMM GUI, OpenEMM creates a new filter address, by default based on the mailloop service domain name. The OpenEMM user should define a mail forwarding for the address(es) used as sender address in its mailings, to direct all incoming response to the filter address(es) for further processing by the mailloop service (see “Alternative A” below).

In our example below the subdomain of the OpenEMM server hostname is *openemm* and the mailloop service subdomain name will be *mailloop*. The (abbreviated) DNS entry for *domain.com* should look like this:

```
openemm                IN    A     0     <ip address>
mailloop.domain.com.   IN    MX    10    openemm.domain.com.
```

Replace expression *<ip address>* with the IP address of the OpenEMM server, which also runs the mailloop service.

The first line assigns the IP address to the regular hostname, and the second line defines the MX record for the mailloop service domain name, meaning that host *openemm* accepts emails sent to host *mailloop*.

Validate your setup by using a tool like *dig* or *host*, for example

```
# host -t A openemm.domain.com
# host -t MX mailloop.domain.com
```

10.1.1 Bounce Filtering Alternatives

When you send emails and want to take advantage of the bounce management for asynchronous bounces, there are three possibilities to set up your bounce management filtering. This chapter explains the different options you have.

The *sender address* mentioned in the options below is the email address used as the sender address in your mailing or newsletter, i.e. the "From:" address (and it is also used as the "envelope from" address, if not explicitly configured otherwise).

© 2023 AGNITAS AG

The *filter address* is the email address hosted by the mailloop service, and it is configured in the GUI of OpenEMM (feature "bounce filter"). The default filter address is created as "reply_<unique number>" in the local part and with the domain name of the mailloop service. In the example below the domain name is "mailloop.domain.com", i.e. the standard filter address created at first would be reply_1@mailloop.domain.com.

The popular option (alternative A):

Use whatever sender address you like, example: *news@agnitas.com*. Create a filter address by setting up a bounce filter in OpenEMM (see user manual). This filter will auto-generate a filter address like reply_1@mailloop.domain.com.

Implement a forward mechanism in the email account of the sender address to forward incoming mail, which was sent back to the sender address, to the filter address of OpenEMM. If you use news@agnitas.com as sender address, set the forwarding for domain agnitas.com with

```
# echo "news: reply_1@mailloop.domain.com" >> /etc/aliases
# newaliases
```

The flow of automated responses to your mailings (like autoresponder mails and bounce notifications) starts at the email address of the recipient, goes back from there to your sender address and is forwarded from the sender address to the filter address.

The advantage of this model is that you can choose any domain for the sender address you want, but you have to implement an external forward mechanism.

The advanced option (alternative B):

Use a sender address with the domain name of the mailloop service (for example news@mailloop.domain.com). Since no real email addresses exist for this sender domain name, normally it would not be possible to reply to an email with this sender address. To forward responses to a valid email address you have to define a bounce filter:

In the GUI configuration for the bounce filter set field *Filter address* manually to your individual sender address (in this example news@mailloop.domain.com) and this address is then bound to this bounce configuration. Due to performance reasons it may take a few minutes until a newly created entry is known by the system.

The flow of automated responses to your mailings (like autoresponder mails and bounce notifications) now goes directly from the email address of the recipient to your sender address, which is equal to the filter address of the bounce filter.

The advantage of this alternative is, that no external forward mechanism is needed, but your sender address has to use the domain name of your mailloop service.

The sophisticated option (alternative C):

There is a third alternative for your filter configuration which allows you to set up a filter address of your own choice without having to implement an external forward mechanism. In this case you create a bounce filter with an individual filter address of your choice, like reply@yourdomain.com. Make sure that in the DNS record of the domain used in your filter address, the MX record points to your OpenEMM server like

```
yourdomain.com. IN    MX    10    openemm.domain.com.
```


© 2023 AGNITAS AG

Now you can use this filter address in a similar way as in alternative B, but with a different domain name (in this example: yourdomain.com). If you use SPF, make sure that the SPF record of this domain used in your filter address includes the domain of your OpenEMM server.

In this case the flow of automated responses to your mailings (like autoresponder mails and bounce notifications) goes directly from the email address of the recipient to your sender address, which is equal to the filter address of the bounce filter.

The advantage of this alternative is, that no external forward mechanism is needed like in alternative A and that the domain of the sender address and the filter address can differ opposed to alternative B.

Caveat: Alternatives B or C will not accept some reserved local parts as part of the filter address. These reserved local parts are currently

- reply_<number>
- aml_<number>
- fbl

10.2 Configuration of Default Settings

Menu *Configuration*, sub-menu *Change configuration in DB* allows you to change certain configuration parameters of the OpenEMM frontend like file paths, server addresses and limiting values. You should have a look at the list of parameters to understand which parameters can be changed.

In the section of parameters with prefix *mailaddress* you should define email addresses for support requests by your users and email addresses for certain notification mails. By default, these email addresses are set to (invalid) sender domain *example.com* and, therefore, would never leave the OpenEMM server.

Some parameters of interest can be modified directly in database table *company_info_tbl*. Parameters with prefix *max* limit certain resources to avoid an overload of the database and parameters with prefix *expire* define, after how many days certain entries are deleted from the database to limit the required storage space.

Table c*onfig_tbl* holds additional parameters valid for the instance of OpenEMM, especially URLs and file paths to components used by OpenEMM as well as system email addresses.

If the OpenEMM database holds more than 10,000 recipients and you open the recipient list you will be greeted with message *The option you selected is too large to be displayed completely. Please limit your selection to reduce the amount of recipients.*

If you want more than 10,000 recipients to be processed for the recipient list (which will take longer to display), set field *max_recipients* in database table *company_tbl* to the value you want:

```
SQL> UPDATE company_tbl SET max_recipients = 100000;
```


© 2023 AGNITAS AG

To increase security, OpenEMM blocks logins when the same IP address generates a certain number of failed logins. The default value for the maximum number of failed logins is 10 and the default value for the lock out time is 60 seconds. You can change both values in the database with a new entry in table *config_tbl*. Examples for max. 3 retries and 5 minutes lock time:

```
SQL> INSERT INTO config_tbl (class, name, value, creation_date,
change_date) VALUES ('loginTracking', 'webui.maxFails', '3',
current_timestamp, current_timestamp);
```

```
SQL> INSERT INTO config_tbl (class, name, value, creation_date,
change_date) VALUES ('loginTracking', 'webui.ipBlockTimeSeconds', '300',
current_timestamp, current_timestamp);
```

If you are really into it, have a look at the source code of class *ConfigValue.java* to see, what else you can configure in OpenEMM.

If any change to the database configuration of OpenEMM does not come into effect within 5 minutes, you have to restart OpenEMM.

10.3 Configuration for MariaDB/MySQL Database

Please be aware that the default value of MariaDB/MySQL parameter *max_allowed_packet* may only be 1 or 2 MByte. In this case, you can not load a single data packet (a file for example) bigger than 1 or 2 MByte into the database. This affects the upload of attachments for emails.

To change the maximum size for these files to a bigger value, you need to set the parameter *max_allowed_packet* in section *[mysqld]* of MySQL's or MariaDB's configuration file (usually *my.cnf* in directory */etc* or */etc/opt/mariadb10x*) to something like the following:

```
max_allowed_packet=10M
```

and restart MariaDB/MySQL afterwards, with

```
# systemctl restart mysql
```

Since the transfer of data to the database has some overhead, the value for *max_allowed_packet* should be a little bit higher than the value for *attachment.maxSize* in file *emm.properties*. You can check the current value of *max_allowed_packet* in MariaDB/MySQL with statement

```
SQL> SELECT @@max_allowed_packet;
```

The value of *max_allowed_packet* also limits the maximum size of SQL statements. In OpenEMM this is important for those SQL statements that retrieve statistical data for display in the workflow manager and dashboard calendar. In (the very unlikely) case that the GUI service does not show these numbers and you find a corresponding error message in Tomcat's log *catalina.out*, double the value for parameter *max_allowed_packet* until it works.

If you want to create emails with emojis, the default collation of MariaDB/MySQL can be a problem. Therefore, change the default collation in file *my.cnf* in section *[mysqld]* to

© 2023 AGNITAS AG

```
character-set-server = utf8mb4
collation-server = utf8mb4_unicode_ci
```

Please restart MariaDB/MySQL after this change.

10.4 Configuration of Webservices

The webservice interface runs as a separate web application in directory */home/openemm/webapps/webservices*.

After OpenEMM has been launched you may request the WSDL file for the webservices via URL

```
http://<domain>/2.0/emmservices.wsdl
```

To be able to access the webservices of OpenEMM you have to create a webservice user with a password first. See the user manual for details.

10.5 Configuration of DKIM Support

Starting with version 21.04, OpenEMM supports DKIM. DKIM configuration and administration is done by two shell scripts in directory */home/openemm/bin*. *dkim-creat* lets you create and implement new DKIM keys. It uses tool *dkim-mgr* for that purpose. *dkim-mgr* is the core tool to maintain DKIM keys.

To create a new DKIM key, launch *dkim-creat* as user *openemm* in your working directory with 3 parameters:

- the domain for which a DKIM key should be created
- the selector to identify the correct DKIM key (like "openemm")
- the bit length of the key (we recommend 2.048 bits, maximum should be 4.096 bits)

dkim-creat generates three files in your current directory:

- *<domain>-pub*, containing the public key
- *<domain>-priv*, containing the private key
- *install-<selector>@<domain>.sh*, a little shell script to save your private DKIM key in the OpenEMM database.

Furthermore, *dkim-creat* shows the necessary configuration of the DKIM entry with the public DKIM key in the DNS record of your sender domain.

This is an example for executing *dkim-creat*:

```
$ dkim-creat sample.com selector 2048
Creating private/public key pair
selector@sample.com.priv/selector@sample.com.pub for sample.com (selector
selector) with 2048 bit
Installation sample for your DNS:
```


© 2023 AGNITAS AG

```
selector._domainkey.sample.com. IN TXT "v=DKIM1;
p=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtWA2bIcq3n95h7hixhTDdSt2Bjc
y209N7OOAHm/81lSY/
PbkY7n5x5LELGvMuuOfg7QVChTM5dslDmJ2EHG8TSptZRuOPCw1fW2X3J8zkJK74wLSSVUoKW
TYwefp7+WYfoY+XxLzc40DWS1kxYNSvO9KaKxctMYYiKruAtdpnDBuK/
tEbTJZBBWH55vwTmXhYbX6L2ZsKIyKjueGfx7RTm3GrVAWRTpBo9krBbM0PL4dP8h3yySr9Hv
/WwXk19qeaC0oksVKChiXpc8CCt1gdLWQ5FeqoKnc6LkMkl4th0gPY/
voJB6EQA1BC5ZNbpYxgma1UEv1bG37iKV7hJ+LhQIDAQAB"
Creating install-selector@sample.com.sh .. add done.
```

As soon as the public DKIM key is filed in the DKIM entry of the DNS record for your sender domain, execute script *install-<selector>@<domain>.sh* to save the private DKIM key into the OpenEMM database.

© 2023 AGNITAS AG

11 OpenEMM Administration

11.1 Automated Startup

If you want OpenEMM to automatically launch at server reboot, you can use a systemd unit file for that purpose. Create a new file *openemm.service* in directory */etc/systemd/system/* with the following content (please note that the *After* properties belong into one line):

```
[Unit]
Description=OpenEMM startup script
After=var-run.mount network.target local-fs.target time-sync.target
postfix.service

[Service]
User=openemm
Group=openemm
Type=oneshot
RemainAfterExit=true
LimitNOFILE=16384
ExecStart=/home/openemm/bin/openemm.sh start
ExecStop=/home/openemm/bin/openemm.sh stop
TimeoutSec=300
StandardOutput=journal+console

[Install]
WantedBy=multi-user.target
```

After deploying this file, reload the systemd-daemon and enable the openemm service with

```
# systemctl daemon-reload
# systemctl enable openemm
```

At next server reboot, OpenEMM will be started automatically.

11.2 Jobqueue Monitoring

The jobqueue is part of the OpenEMM frontend and handles a lot of automated OpenEMM processes (jobs). It makes sense to check the status of the jobqueue and its jobs from time to time, because some hanging jobs like cleanup processes will have no immediate effects but can drag down performance of OpenEMM over time.

You can check the status of the jobqueue in the GUI of OpenEMM in menu *Administration*, sub-menu *System Status*. This page shows the overall status of the jobqueue, while tab *Show Jobqueue* lists all available jobs and its individual status.

© 2023 AGNITAS AG

If you do not want to check the status of OpenEMM jobs in the frontend, you can also access the OpenEMM database directly. The SQL statement is

```
SQL> SELECT id, description, lastresult, running, nextstart, laststart
FROM Job_queue_tbl
WHERE deleted = 0 AND
( (lastresult != 'Ok') -- has errors
  OR (nextstart < CURRENT_TIMESTAMP() - INTERVAL 10 MINUTE) -- overdue
  OR ( (running = 1) AND
    (laststart < CURRENT_TIMESTAMP() - INTERVAL 4 HOUR)) -- (too) long running
);
```

To automate the jobqueue check, you may run this statement from a shell script and analyze its ouput with a monitoring software like Icinga.

11.3 Database Backup

For MariaDB/MySQL there exist plenty of strategies for database backups and tons of books and Internet resources on that subject. However, if you run only a medium MariaDB/MySQL database with a few GByte of data and if you can live with an interruption of services of very few minutes, you may simply use tool *mysqldump*:

```
# mysqldump -aCceQx --hex-blob --routines --triggers -u root -p -r
openemm.sql openemm
```

Executed at the command line, this statement copies a database dump in a very robust format into text file *openemm.sql*. The database dump can be imported back into an emtpy database *openemm* simply with

```
# mysql -u root -p openemm < openemm.sql
```

Menu *Database Maintenance* of *OMT* offers a backup and restore of the OpenEMM database based on these commands.

If you run a bigger MariaDB/MySQL database which should not be stopped during backup time, we recommend the use of the tool *Percona XtraBackup*.

11.4 Generic Database Tuning

80% of all application performance problems are really database performance problems. If you run a big OpenEMM installation and you are not satisfied with the application's performance, here are some database tuning tips you should try.

If certain tenants of your OpenEMM database hold a long list of recipients, you may speed up database operations like calculating statistics significantly with a combined index on several fields of table *customer_1_binding_tbl.*

We recommend the following two indices in case they do not exist yet:

```
SQL> CREATE INDEX custbind1$mlid_user_cuid$idx ON customer_1_binding_tbl
(mailinglist_id, user_status, customer_id);
```


© 2023 AGNITAS AG

```
SQL> CREATE INDEX custbind1$user_mlid_cuid$idx ON customer_1_binding_tbl
(user_status, user_type, mailinglist_id, customer_id);
```

If you use any other profile field than *email* for duplicate checks, you should put an index on this field in *customer_1_tbl*:

```
SQL> CREATE INDEX cust1$<fieldname>$idx ON customer_1_tbl (<fieldname>) ;
```

Replace placeholder *<fieldname>* with the name of the database field you use for duplicate checks.

11.5 MariaDB/MySQL Database Tuning

Since version 5.5 of MySQL, InnoDB has been the default engine of MySQL (and later MariaDB). While InnoDB supports row locking and real transactions for better crash protection (opposed to MyISAM), the internal data structure is more complex than MyISAM's, which leads to larger table sizes, slower writes, slower full table scans and slower handling of BLOBs and CLOBs. Also, backup and recovery via *mysqldump/mysql* is slower.

Because InnoDB is much more sensitive to configuration parameters than MyISAM, you should at least add properties *innodb_buffer_pool_size* and *innodb_log_file_size* in section *[mysqld]* of MySQL's or MariaDB's configuration file (usually *my.cnf* in directory */etc* or */etc/opt/rh/rh-mariadb10x*), because the internal default values of 128 MByte and 5 MByte are much too small for bigger databases with lots of InnoDB tables.

As a rule of thumb: *innodb_buffer_pool_size* should be set to 50% of the RAM of your server and *innodb_log_file_size* should be set to ¼ of the size of *innodb_buffer_pool_size*, but not higher than 256 MByte to limit recovery time after a database crash.

To prevent the InnoDB engine from saving all table data into system tablespace file *ibdata1* in directory */var/lib/mysql* you may add property

```
innodb_file_per_table=1
```

in section *[mysqld]*.

If you would like to get some more recommendations on how to optimize MySQL's or MariaDB's configuration file, we suggest to run the old but trusted *tuning-primer* script available at http://www.day32.com/MySQL/tuning-primer.sh

11.6 Stopping the Sending in Case of Emergency

A mail dispatach can be stopped in the GUI of OpenEMM. But if you want to dig deeper and stop the sending of a certain mailing at the command line level, it is important to understand the process of its generation, distribution and dispatch:

Meta mail packages are generated by backend. These files are located in directory

© 2023 AGNITAS AG

```
/home/openemm/var/spool/META
```

and one part of the file name is the ID of the mailing. If files with the ID of the mailing to be stopped, are removed, they are no longer distributed by the backend for dispatch to its mailer service. However, to avoid internal conflicts, you have to stop process *pickdist* before deleting the files with

```
# /home/openemm/bin/pickdist.sh stop
```

and you have to re-start *pickdist* afterwards with

```
# /home/openemm/bin/pickdist.sh start
```

If the backend has already distributed meta mail packages for processing by its mailer service, you will find the packages which have not been processed yet or which are in process right now in directory

```
/home/openemm/var/spool/mail/incoming
```

If a file is already in process, deleting it does not help. You may check if a file is in process with

```
# ps aux | grep xmlback
```

If you see process *xmlback* with a file name as argument, this file is already in process, i.e. process *xmlback* generates the final email files out of the meta mail package.

Otherwise you may delete the file(s). In this case you should stop process *pickdist* before deleting the files with

```
# /home/openemm/bin/pickdist.sh stop
```

and you have to re-start *pickdist* afterwards with

```
# /home/openemm/bin/pickdist.sh start
```

If a meta mail package is in process by the mailer service, there is no easy way to delete the email files generated by these packages. You may stop MTA Postfix, check the mail queues and manually delete the files of all emails belonging to the mailing to be stopped.

Please be aware that the statistics for a certain mailing in the GUI of OpenEMM may not be correct if you manually delete meta mail packages of this mailing or email files of this mailing from the MTA queues.

11.7 Out of Memory

If you work with big lists and experience an error message like

```
Java.lang.OutOfMemoryError: Java heap space
```

you have to allocate more memory to the Java Virtual machine (JVM). You can increase the minimum and maximum memory in file *emm.sh.additional.properties* (which overwrites settings of *emm.sh*) in directory */home/console/bin* or */home/rdir/bin* by increasing the values of parameters **-Xms** for minimum and **-Xmx** for maximum memory in variable *JAVA_OPTS_EXTERNAL*. If you have allocated all memory available and the error remains, you should increase your server RAM to at least 2 GByte (better: 4 GByte) and modify the parameters accordingly.

© 2023 AGNITAS AG

11.8 Log Rotation

To prevent the Tomcat log from filling up the hard disk of the OpenEMM server, you may install a log rotation to get rid of old log files. Create file *tomcat-openemm* in directory */etc/logrotate.d* with this suggested content:

```
/home/openemm/logs/catalina.out {
  copytruncate
  daily
  rotate 7
  compress
  dateext
  size 10k
  missingok
  sharedscripts
  postrotate
    ########################################################################
    # zip files older than 180 min and delete access_logs older than 90 days
    ########################################################################
    find /home/openemm/logs/access -name "*.log" -mmin +180 -exec gzip -9 {} \;
    find /home/openemm/logs/access -name "*.log.gz" -mtime +90 -exec rm {} \;
    find /home/openemm/logs -name "*.gz" -mtime +10 -exec rm {} \;
    find /home/openemm/logs -name "*manager*.log" -mmin +180 -exec rm {} \;
    find /home/openemm/logs -name "localhost*.log" -mmin +180 -exec rm {} \;

    ################################################
    # delete files older than 5 days in logs/webapps/
    ################################################
    find /home/openemm/logs/webapps -type f -mtime +5 -exec rm {} \;

    ###############################
    # Delete old ARCHIVES > 10 days
    ###############################
    find /home/openemm/var/spool/ARCHIVE -type d -mtime +10 2>/dev/null | xargs
-r rm -fr

    ###################################
    # Delete old logfiles from backend
    ###################################
    find /home/openemm/var/log -name "*.log" -mtime +30 -exec rm {} \;
  endscript
}
```

For redirect servers you should replace value *10k* in line 6 to *1M* and replace *console* with *rdir* in all paths. You also may shorten the various retention times to your individual needs.

© 2023 AGNITAS AG

11.9 Changing Security-Related Files

OpenEMM passwords are saved in the database not only encrypted but salted as well. The salt file with the file extension *salt* is located in directory */home/openemm/conf/keys*.

If you want to change the salt file, please do it before you start operating OpenEMM, because otherwise all saved passwords will not work any longer. For generating a new salt just save a string of letters, digits and other characters of the ASCII character set (decimal values 33 to 126) with a maximum length of 32 characters in a simple text file and name the new file like the old salt file. You may change the file name in configuration file *emm.properties* of the corresponding service.

To prevent access of the statistics service by a random server, the statistics service is accessible via HTTPS protocol only, uses a private key and grants access only to those servers providing the corresponding public key. This public key is provided by the GUI service. The private key in file *birt_private.pem* is located in directory */home/openemm/conf/keys/*. The corresponding public key in file *birt_public.pem* is located in the same directory, since in OpenEMM GUI and statistics service operate on the same server.

If you think that the keys are not safe (enough) for your purpose, you may replace them and restart OpenEMM.

11.10 Switching SMTP Server from Sendmail to Postfix

If you want to switch the type of SMTP server used by OpenEMM from Sendmail to Postfix after OpenEMM has gone productive, you should wait patiently until all pending mailings have been delivered and stop OpenEMM with

```
# openemm.sh stop
```

As second step you have to stop the current SMTP server:

```
# systemctl stop sendmail
```

Next step should be to clean the mail queues of the current SMTP server to avoid sending out old mails in case you will switch back later. To clean Sendmail's mail queues execute

```
# rm -r /home/mailout/*QUEUE*/*
```

Before you switch to the new default SMTP server, make sure that you have installed the required packages and that you have made the required configuration modifications explained in section *Postfix Deployment*.

You have to switch the default SMTP server to Postfix with

```
# alternatives --set mta /usr/sbin/sendmail.postfix
```

After that you may launch the new SMTP server with

```
# systemctl start postfix
```


© 2023 AGNITAS AG

and finally restart OpenEMM:

```
# openemm.sh start
```

You may check the type of SMTP server being active on a server running a mailer service with

```
# su - -c 'source scripts/config.sh; echo $MTA' openemm
```


© 2023 AGNITAS AG

12 Apache Native Library

12.1.1 HTTPS for Tomcat

While it is technically possible to access OpenEMM with the HTTP protocol, this is certainly not recommended for production environments. One could even argue that is illegal in EU countries where the GDPR is in force. To use OpenEMM with secure HTTPS connections, for Tomcat you should use the NIO Connector, because according to the Apache Foundation the APR/Native HTTP Connector is deprecated and will be removed in Tomcat 10.1 onwards.

Configuration of the NIO Connector is done by changing the connector type and its properties for Tomcat in tag *Connector* of Tomcat's server configuration file *server.xml* like this:

```
<Connector
  port="8443"
  protocol="org.apache.coyote.http11.Http11NioProtocol"
  scheme="https"
  secure="true"
  SSLEnabled="true"
  disableUploadTimeout="true"
  acceptCount="100"
  connectionTimeout="20000"
  maxThreads="1000"
  enableLookups="false"
  useBodyEncodingForURI="true"
  server="<server_name>">
  <UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />
  <SSLHostConfig
    disableCompression="true"
    honorCipherOrder="true"
    ciphers="ECDHE-ECDSA-AES256-GCM-SHA384 ECDHE-RSA-AES256-GCM-SHA384
ECDHE-ECDSA-CHACHA20-POLY1305 ECDHE-RSA-CHACHA20-POLY1305 ECDHE-ECDSA-
AES128-GCM-SHA256 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-ECDSA-AES256-SHA384
ECDHE-RSA-AES256-SHA384 ECDHE-ECDSA-AES128-SHA256 ECDHE-RSA-AES128-
SHA256"
    protocols="all">
    <Certificate
      certificateChainFile="<tls-ca-keychain-bundle.file>"
      certificateFile="<tls-certificate.file>"
      certificateKeyFile="<tls-private-key.file>" />
  </SSLHostConfig>
</Connector>
```

Replace placeholder *<server_name>* with the FQDN of your OpenEMM server. Replace *<tls-ca-keychain-bundle.file>* with the path and name of your *cacert* file. Replace *<tls-*

© 2023 AGNITAS AG

certificate.file> with the path and name of your *crt* file. And replace <*tls-private-key.file*> with the path and name of your *key* file.

Additionally, you should remove line

```
<Listener
className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
```

from file *server.xml* in case it exists.

The modified *server.xml* file belongs into Tomcat's configuration directory */home/openemm/conf*. Do not forget to activate a port forwarding from port 443 to 8443, because Tomcat uses port 8443 for HTTPS traffic by default.

© 2023 AGNITAS AG

Glossary

12.2 Bounce Management

OpeEMM's automated bounce management provides you with the capability to keep your mailing lists clean and up-to-date automatically. A bounce message is an error message, which is sent from a mail server on the recipient's side to the sender if an email is not deliverable. Bounce management administers emails which are undeliverable temporarily (soft bounce) or permanently (hard bounce). It also filters error messages and autoresponder mails.

OpenEMM does not treat all hardbounce messages reported by remote mail servers as hardbounce. In fact, some messages are only treated as softbounces, although their bounce codes starting with 5 would indicate a hardbounce.

The reason for this kind of ignorant behaviour ist intentional, because some mail servers are not properly configured regarding the generation of hardbounce messages and mistakenly report permanent delivery errors - some even by intention to pretend that certain email addresses do not exist. If OpenEMM would handle those fake hardbounce messages as real hardbounces email addresses of existing recipients would be disabled. As result, we only try to accept bounces as hardbounces which are really proved to be hardbounces.

If you want to modify the bounce management of OpenEMM, file *bav.rule* in directory */home/openemm/lib* is the right place. This file lists in section *[hard]* bounce messages which are recognized as hardbounces, and section *[soft]* lists bounce messages recognized as softbounces. The messages are formatted as regular expressions to allow, among others, the use of wildcards. You may add your own set of messages here.

By the way, if a hardbounce message is recognized as a softbounce even if it is a real hardbounce, this is not a problem. Because a real hardbounce is reported for each mailing again and is counted as a softbounce each time, it will be finally caught by the softbounce scoring of OpenEMM and converted to a hardbounce in the end.

12.3 DNS

DNS is the abbreviation for Domain Name System. This system forwards requests directed to a FQDN to a certain IP address. Each entry in the DNS maps the IP address of a server to a human readable address. Example: In place of the IP address 83.169.23.100, which points to the AGNITAS website, you may use the DNS address *www.agnitas.com*, which is much more convenient (for a human).

© 2023 AGNITAS AG

12.4 FQDN

A Fully Qualified Domain Name (FQDN) links to an IP address of a server. The FQDN may be composed of letters and numbers and by using this option nobody has to remember the difficult number sequence (IP). A FQDN is divided in three levels:

- The affix of the domain is the Top Level Domain (TLD). Example: *com*, *org* or *de*
- The domain name will be inserted in front of the TLD. Example: *agnitas*
- The FQDN starts with the *hostname*. For webpages this is very often *www*

Example: The FQDN *www.yourdomain.com* is composed of

- *www* = hostname
- *yourdomain* = domain name
- *com* = TLD

As you can see, the FQDN consists of the hostname, the domain name and the top level domain separated by dots. The combination of domain name and TLD is commonly referred as *domain*. The FQDN can be expanded by a *subdomain* (like *miami*). The subdomain will be inserted between the hostname and the domain. Example: www.miami.yourdomain.com .

12.5 Softbounce Scoring

If an email address generates lots of softbounces (temporary delivery problems) this is actually an indication that the email address is undeliverable permanently (hardbounce). OpenEMM provides softbounce scoring to identify those email addresses and to convert them to hardbounces.

The rules for converting a softbounce to a hardbounce work like this:

1. Select all email addresses in the softbounce table which generated more than 40 softbounces and where the time-lag between the first and last bounce is longer than 30 days.
2. If no mail opening or link click was registered within the last 30 days for an email address which matchs the before-mentioned conditions, this address is flagged as a hardbounce.
3. If at least one opening or click was registered within the last 30 days, this address is removed from the softbounce table, i.e. its bounce count is reset to zero.

© 2023 AGNITAS AG

Third Party Licenses

OpenEMM relies on several great open source frameworks, libraries and tools. In order to give due credit to those fine projects, you will find the individual licenses of the open source projects used by OpenEMM in subdirectory */home/openemm/webapps/emm/licences*.

© 2023 AGNITAS AG