# Linux - Installing or Upgrading (1.3.2)


- Requirements
- Description
- Installation on a new server
- Upgrade on an existing server
- Backup an existing installation
- Restore a backup
- Uninstalling
- Checking Dependencies
- Installing Dependencies
    - Installing Dependencies for RedHat 6 / Centos 6 servers
    - Installing Dependencies for Debian / Ubuntu servers


## Requirements

Root level access to the Linux server.

Basic Linux knowledge.

A 64bit linux server running one of - RedHat 6, Centos 6, Debian 6 or 7, Ubuntu 12.04 or 14.04.

## Description

As at version 1.3.1 of Open-AudIT on 16th May, 2014 (for RedHat / Centos installs only at this stage), an installation script is now shipped in the tarball. This script can perform installs, upgrades, backups, restores, uninstalls and check for install dependencies. How to perform all of the above is detailed below. The script will log it's commands to the file /tmp/install.log. If your install fails, this log will provide valuable information as to why.

NOTE - Any commands that have <SOMETHING> in them require that you substitute <SOMETHING> for the appropriate value. For example, if your server's hostname is 'vali', and the command states *echo "<HOSTNAME>" >> somefile.txt* you should type *echo "vali" >> somefile.txt*. The specific value of <HOSTNAME> refers to the hostname of your server.

All commands should be run as root (you can "sudo su" on Ubuntu). All steps below require the tarball to be copied to /tmp, extracted and then the install script (install.pl) run from that directory.

```
cd /tmp
tar xf OAE-Linux-x68_64-release_1.3.1.tar.gz
```

## Installation on a new server

To install Open-AudIT on a new server (which does not have an existing Open-AudIT installation) perform the following steps (after copying and extracting the tarball as above):

Run the script by entering the command

```
./install.pl
```

The script will check for dependencies. If the dependencies are not met, some information about how to install them is displayed and the script will exit. To force the install, regardless of the dependency status, run the command

```
./install.pl check_dependencies=n
```

The script will install the files and prompt for your MySQL root user password.

The database will be setup, the daemon configured and started and the install will be completed.

To access the application, go to http://<HOSTNAME>/omk/oae and you will see a logon screen.

## Upgrade on an existing server

To upgrade an existing installation of Open-AudIT perform the following steps (after copying and extracting the tarball as above):

Run the script by entering the command

```
./install.pl
```

The script will check for dependencies. If the dependencies are not met, some information about how to install them is displayed and the script will exit. To force the install, regardless of the dependency status, run the command

```
./install.pl check_dependencies=n
```

Upon completing dependency checking (or not), the script will ask if you would like a backup of your files and database taken. If you answer 'y', a tarball will be created in /tmp named open-audit_backup-YYYY-MM-DD-HHMM.tar.gz Where YYYY is the year, MM is the month, DD is the day, HH is the hour and MM is the minute. Included in this backup will be the install script itself so a restore is as simple as an install or upgrade. See below for further details.

Your original install folders /usr/local/omk and /usr/local/open-audit will be moved to their original name + the timestamp. The new files will be copied into new folders and any existing attachments will be copied from the old too the new folders.

The new web files will be copied to your web server directory.

Your database will NOT be upgraded. You will need to log on to Open-AudIT - when you do you will be prompted to upgrade it. Go to http://<HOSTNAME> /open-audit/index.php/main/list_groups/0

## Backup an existing installation

To backup an existing installation without installing or upgrading, perform the following steps (after copying and extracting the tarball as above):

Run the script by entering the command

```
./install.pl backup_only=y
```

The script will confirm you wish to take a backup of your existing files and data. If you answer 'y', a tarball will be created in /tmp named open-audit_backup-YYYY-MM-DD-HHMM.tar.gz Where YYYY is the year, MM is the month, DD is the day, HH is the hour and MM is the minute. Included in this backup will be the install script itself so a restore is as simple as an install or upgrade. See below for further details.

## Restore a backup

To restore a backup taken previously by the script, perform the following steps (after copying the backup file to /tmp):

Run the script by entering the command

```
cd /tmp
tar xf open-audit_backup-YYYY-MM-DD-HHMM.tar.gz
./install.pl restore=y
```

You will be prompted for your MySQL root user credentials. Any existing Open-AudIT files and database will be removed from the server (if they exist). The backed-up files will be restored to the server, the daemon installed and configured, the database restored and the daemon started.

## Uninstalling

To uninstall Open-AudIT and delete all data  perform the following steps (after copying and extracting the tarball as above):

Run the script by entering the command

```
./install.pl uninstall=y
```

The script will offer to provide a backup of the existing files and data (as per above) and then confirm that you do wish to uninstall Open-AudIT. If you answer 'y', the files will be deleted, the daemon removed and the database and database user dropped.

## Checking Dependencies

To check the dependencies are installed without actually installing perform the following steps (after copying and extracting the tarball as above):

Run the script by entering the command

```
./install.pl check_dependencies_only=y
```

The script will run, check the dependencies are installed and inform you if they are or provide information on which packages are not and offer the commands for you to install them.

## Installing Dependencies

The check dependencies option above should inform you which packages are missing from your system. To install these packages follow the instructions below as appropriate for your operating system.

### Installing Dependencies for RedHat 6 / Centos 6 servers

The complete list of packages required by a RedHat/Centos install are - mysql, mysql-server, httpd, php, php-cli, php-mysql, php-ldap, php-mbstring, php-mcrypt, php-process, php-snmp, php-xml, nmap, zip, curl, wget, sshpass, screen, samba-client, winexe.

Ensure your package manager is up to date

```
yum update
```

You will need an external repo to install some items. Enable the repo

```
rpm -Uvh http://download.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
```

Install the missing dependencies by copying and pasting the output from the script "yum install package1 package2 etc etc".

If you have not already, download the appropriate 'winexe' package from the repository at http://download.opensuse.org/repositories/home:/ahajda:/winexe/

Install it

```
yum install winexe
```

SELinux is known to cause some issues. Disable it by

```
sed -i -e 's/SELINUX=/#SELINUX=/g' /etc/selinux/config
echo "SELINUX=disabled" >> /etc/selinux/config
setenforce 0
```

Configure IP Tables to allow the Apache traffic

```
sed -i 's/\*filter$/*filter\n-A INPUT -m state --state NEW -p tcp --dport 443 -j ACCEPT/' /etc/sysconfig
/iptables
sed -i 's/\*filter$/*filter\n-A INPUT -m state --state NEW -p tcp --dport 80 -j ACCEPT/' /etc/sysconfig/iptables
/etc/init.d/iptables restart
```

You will need to ensure your PHP timezone is set correctly. You can check which time zones PHP supports at http://www.php.net/manual/en/timezones.php You can find out your server's timezone by

```
cat /etc/sysconfig/clock | grep ZONE | cut -d"\"" -f2
```

Configure PHP (substituting <TIMEZONE> from above). Set your PHP defaults

```
sed -i -e 's/memory_limit/;memory_limit/g' /etc/php.ini
echo "memory_limit = 512M" >> /etc/php.ini
sed -i -e 's/max_execution_time/;max_execution_time/g' /etc/php.ini
echo "max_execution_time = 300" >> /etc/php.ini
sed -i -e 's/max_input_time/;max_input_time/g' /etc/php.ini
echo "max_input_time = 600" >> /etc/php.ini
sed -i -e 's/error_reporting/;error_reporting/g' /etc/php.ini
echo "error_reporting = E_ALL" >> /etc/php.ini
sed -i -e 's/display_errors/;display_errors/g' /etc/php.ini
echo "display_errors = On" >> /etc/php.ini
sed -i -e 's/upload_max_filesize/;upload_max_filesize/g' /etc/php.ini
echo "upload_max_filesize = 10M" >> /etc/php.ini
sed -i -e 's/date.timezone/;date.timezone/g' /etc/php.ini
echo "date.timezone = <TIMEZONE>" >> /etc/php.ini
```

If you have had to install MySQL and Apache, configure and start the daemons

```
chkconfig --levels 235 mysqld on
service mysqld start
echo "ServerName <HOSTNAME>" >> /etc/httpd/conf/httpd.conf
chkconfig --levels 235 httpd on
chsh -s /bin/bash apache
service httpd start
```

Lastly, set the SUID for the nmap binary (so we can use the apache front end to run scripts which call nmap).

NOTE - This command will likely need to be re-run if Nmap is upgraded.

```
chmod u+s /usr/bin/nmap
```

## Installing Dependencies for Debian / Ubuntu servers

The complete list of packages required by a Debian/Ubuntu install are - mysql-server, apache2, libapache2-mod-proxy-html, libapache2-mod-php5, openssh-server, php5, php5-ldap, php5-mcrypt, php5-mysql, php5-snmp, nmap, snmp, zip, wget, curl, sshpass, screen, smbclient, winexe.

Ensure your package manager is up to date

```
apt-get update
```

Install the missing dependencies by copying and pasting the output from the script "apt-get install package1 package2 etc etc".

If you have not already, download the appropriate 'winexe' package from the repository at http://download.opensuse.org/repositories/home:/ahajda:/winexe/

Install it

```
dpkg -i install winexe
```

You will need to ensure your PHP timezone is set correctly. You can check which time zones PHP supports at http://www.php.net/manual/en/timezones.php You can find out your server's timezone by

```
cat /etc/timezone
```

Configure PHP (substituting <TIMEZONE> from above). Set your PHP defaults

```
sed -i -e 's/memory_limit/;memory_limit/g' /etc/php5/apache2/php.ini
echo "memory_limit = 512M" >> /etc/php5/apache2/php.ini
sed -i -e 's/max_execution_time/;max_execution_time/g' /etc/php5/apache2/php.ini
echo "max_execution_time = 300" >> /etc/php5/apache2/php.ini
sed -i -e 's/max_input_time/;max_input_time/g' /etc/php5/apache2/php.ini
echo "max_input_time = 600" >> /etc/php5/apache2/php.ini
sed -i -e 's/error_reporting/;error_reporting/g' /etc/php5/apache2/php.ini
echo "error_reporting = E_ALL" >> /etc/php5/apache2/php.ini
sed -i -e 's/display_errors/;display_errors/g' /etc/php5/apache2/php.ini
echo "display_errors = On" >> /etc/php5/apache2/php.ini
sed -i -e 's/upload_max_filesize/;upload_max_filesize/g' /etc/php5/apache2/php.ini
echo "upload_max_filesize = 10M" >> /etc/php5/apache2/php.ini
sed -i -e 's/date.timezone/;date.timezone/g' /etc/php5/apache2/php.ini
echo "date.timezone = <TIMEZONE>" >> /etc/php5/apache2/php.ini
```

You may need to manually enable mcrypt in PHP.

```
php5enmod mcrypt
```

Set the server name for Apache, enable mod-proxy and restart

```
echo "ServerName <HOSTNAME>" >> /etc/apache2/apache2.conf
a2enmod proxy_http
service apache2 restart
```

Lastly, set the SUID for the nmap binary (so we can use the apache front end to run scripts which call nmap).

```
dpkg-statoverride --update --add root root 4755 /usr/bin/nmap
```