OpenEMR 4.1.1 Linux Installation

Required Software Installation (distribution specific)

First install and configure MySQL, Apache (or another PHP-capable webserver), and PHP5. We recommend referring to your specific operating system (OS) documentation for installation of these software packages.

* + **(Ensure PHP is configured with**[**these settings**](http://www.open-emr.org/wiki/index.php/FAQ#What_are_the_correct_PHP_settings_.28can_be_found_in_the_php.ini_file.29_.3F)**)**

**Debian Lenny**

To install the prerequisite software on debian, use the APT system as the system administrator to install packages listed on following wiki page section:

[OpenEMR Dependencies](http://www.open-emr.org/wiki/index.php/OpenEMR_System_Architecture#OpenEMR_Dependencies)

**Centos Or RedHat**

**If using CentOS or Redhat**, then ensure when configuring MySQL for OpenEMR, STRICT\_MODE must be disabled. This can be done in the MySQL configuration (my.cnf) file. You can check the setting in an existing MySQL instance by executing **SHOW VARIABLES;** in the MySQL Command Line Client and checking the value of SQL\_MODE. For more details see: <http://dev.mysql.com/doc/refman/5.0/en/server-sql-mode.html>)

Installing OpenEMR

Extracting the Tarball

**1.** First, Download OpenEMR from sourceforge:

<http://downloads.sourceforge.net/openemr/openemr-4.1.1.tar.gz> (MD5 sum: 47caa36d37498f5ca9f0e30329798414)

**2.** Next, Extract the downloaded archive:

**tar -pxvzf openemr-4.1.1.tar.gz**

**3.** Move the extracted openemr-4.1.1 directory to your webserver root directory, and then go to this directory. Refer to your OS and webserver documentation to obtain you webserver root directory. For example, Ubuntu's webserver root directory is **/var/www**, so in Ubuntu you would use the following commands:

**mv openemr-4.1.1 /var/www/openemr**

**cd /var/www**

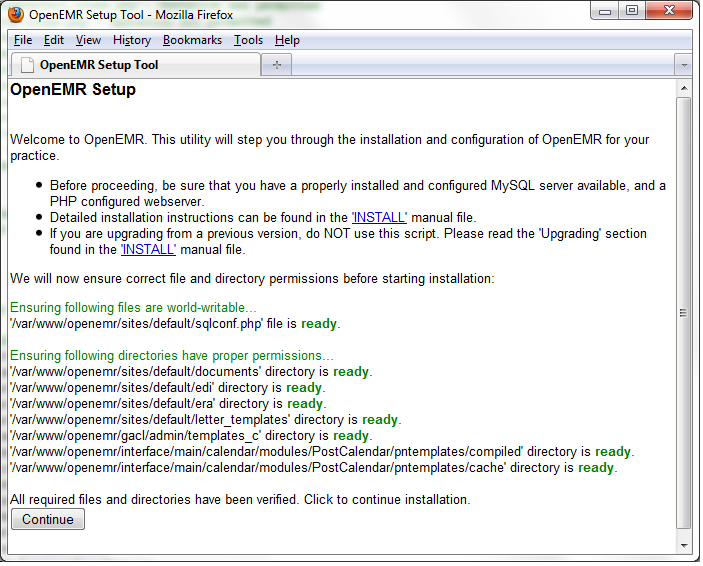
Some other common webserver root directories are:

* + Mandriva: /var/www/html/
  + The FHS: /srv/www/

Configuring the Install through the WEB GUI

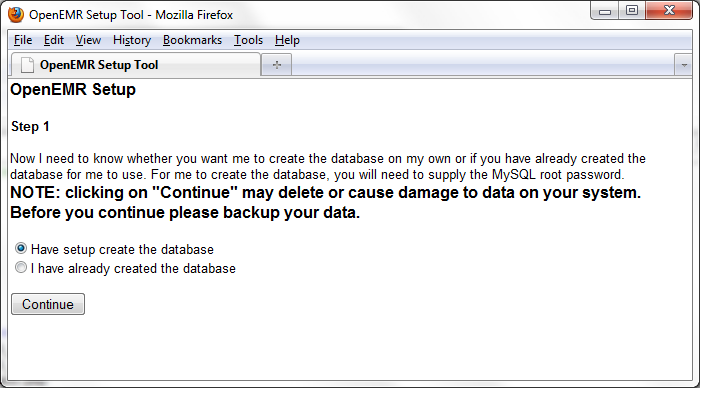
**1.** Open up web-browser and point it to the installation script at <http://localhost/openemr> .

The first screen gives some general instructions and ensures file and directory permissions are set up correctly. It should tell you all your file system permissions are good, and have a 'Continue' button on the bottom of the form. If it does, Click the 'Continue' button. Otherwise, If there are errors on this screen, then follow instructions to fix. (screenshot below)



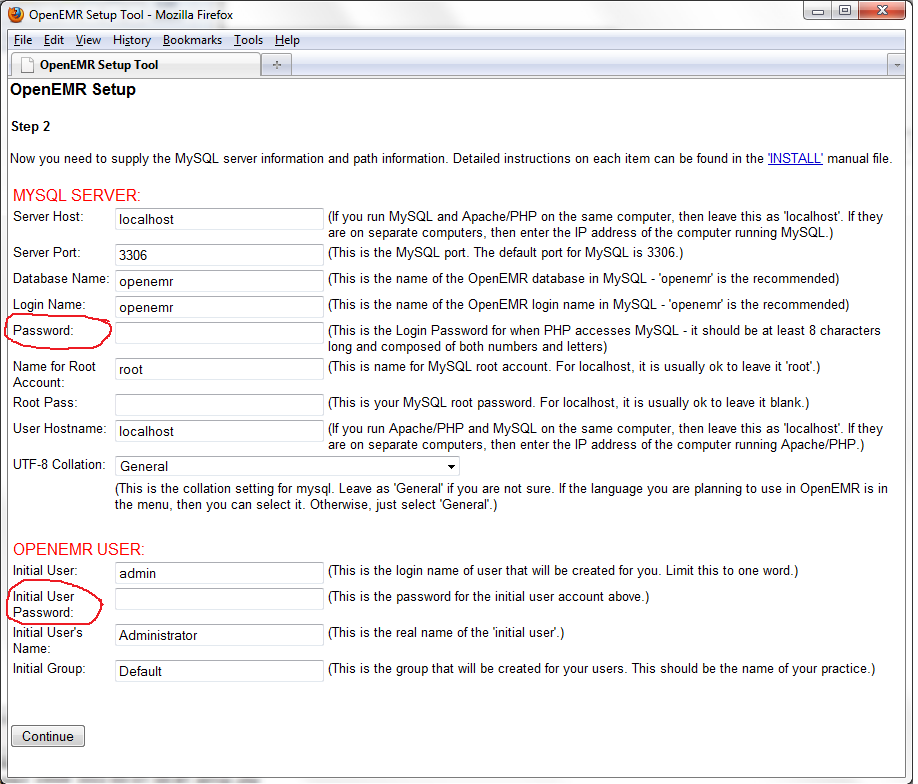
**2.** The next page is 'OpenEMR Setup' 'Step 1', which asks whether setup should create the database.

Leave the setting set to 'Have setup create the database', and click the 'Continue' button. (screenshot below)



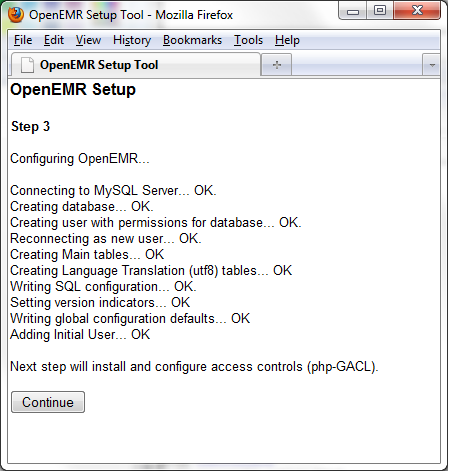
**3.** The next page is 'OpenEMR Setup' 'Step 2'. It has a list of options used to setup the database, along with OpenEMR user and OpenEMR path information.

Most users will just need to enter a 'Initial User Password' for the OpenEMR User and a 'Password' for the sql user(these fields are circled in red in below screenshot). Some users may also have to enter the password of the 'root' user in the mysql server. (screenshot below)



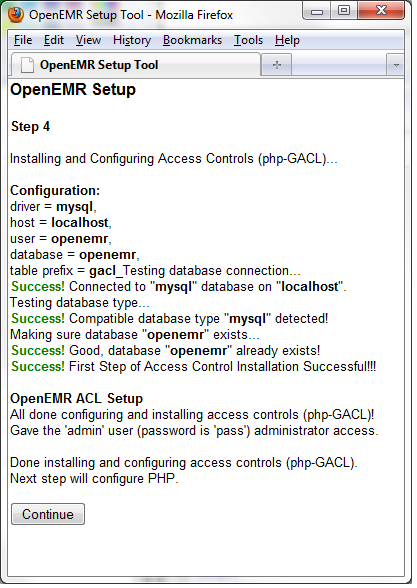
**4.** The next page is 'OpenEMR Setup' 'Step 3'. On this page you should see logs of OpenEMR connecting and creating the database.

At the bottom of the page, it should read 'Next step will install and configure access controls (php-GACL).' Click the 'Continue' button. (screenshot below)



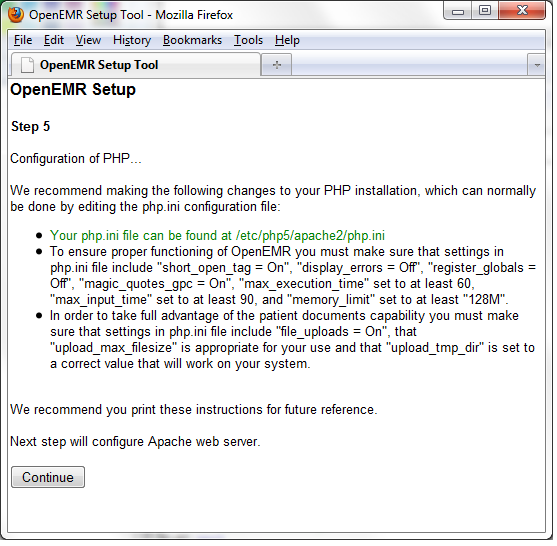
**5.** The next page is 'OpenEMR Setup' 'Step 4'. On this page, OpenEMR configures access controls (php-GACL).

You should see logs of OpenEMR installing and configuring php-GACL. you should see 'Success!' in green lettering four times. At the bottom of the page, it should read 'Next step will configure PHP.'. Click the 'Continue' button. (screenshot below)



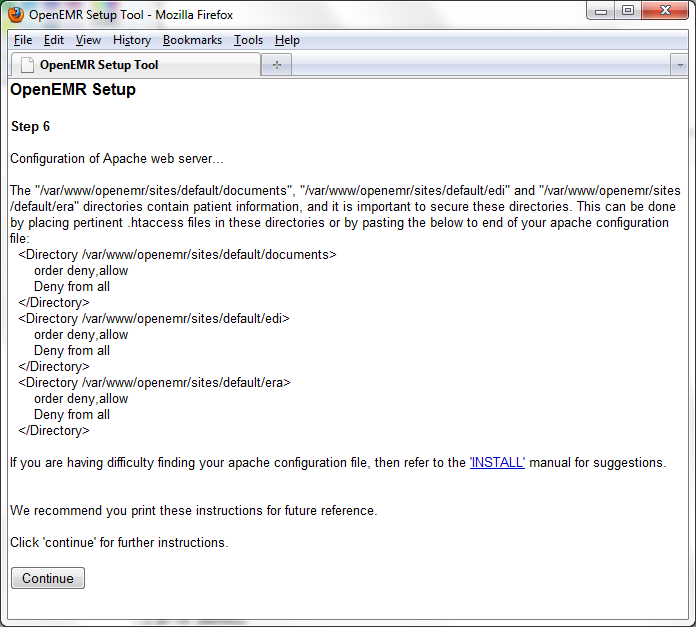
**6.** The next page is 'OpenEMR Setup' 'Step 5'. On this page is displayed recommendations of how to configure your php.ini config file.

Read and follow the instructions carefully. At the bottom of the page, it should read "Next step will configure Apache web server.". We suggest you print this screen for future reference. Click the 'Continue' button. (screenshot below)



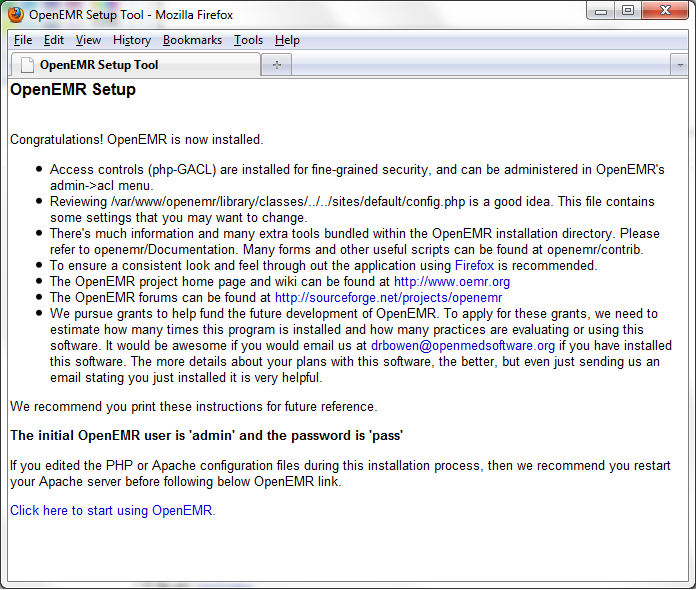
**7.** The next page is 'OpenEMR Setup' 'Step 6'. On this page is displayed recommendations and warnings regarding directories Apache should NOT allow the outside world to see, as these directories contain Protected Health Information.

Read and follow the instructions carefully. We suggest you print this screen for future reference. Click the 'Continue' button. (screenshot below)

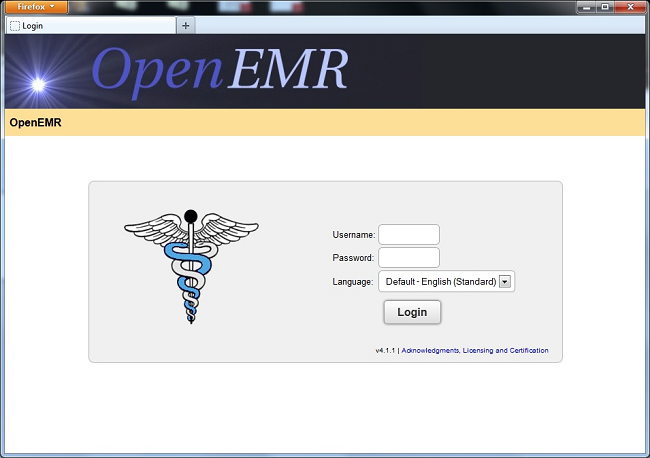


**8.** The next page is 'OpenEMR Setup'. This is the final configuration page.

This final installation screen gives additional instructions. This information is personalized for your setup, so we suggest you print this screen for future reference. At the bottom of the page is your OpenEMR username and password, along with a link to OpenEMR. (screenshot below)



OpenEMR after installation: (screenshot below)



Post-install FileSystem Permission Cleanups

**1.** After installation is finished, then secure files (this should be done as root user):

**chmod 644 openemr/library/sqlconf.php**

**chmod 600 openemr/acl\_setup.php**

**chmod 600 openemr/acl\_upgrade.php**

**chmod 600 openemr/sl\_convert.php**

**chmod 600 openemr/setup.php**

**chmod 600 openemr/sql\_upgrade.php**

**chmod 600 openemr/gacl/setup.php**

**chmod 600 openemr/ippf\_upgrade.php**

Download and install most recent patch

Instruction to do this can be found on the [OpenEMR Patches](http://www.open-emr.org/wiki/index.php/OpenEMR_Patches) page.

Using OpenEMR

See the [OpenEMR 4.1.1 User Guide](http://www.open-emr.org/wiki/index.php/OpenEMR_4.1.1_Users_Guide)

Securing OpenEMR

Overview

With the advent of the Patient Portals, the community is now addressing the issues of how to best secure OpenEMR instances that are open to the web. Doing this requires a firm understanding of securing several parts, which at least include OpenEMR itself, Apache, MySQL, PHP, firewall, router, https, certficates, etc. A forum that began to discuss this issue can be [found here](http://sourceforge.net/projects/openemr/forums/forum/202505/topic/4769926).

There is also a very nice write up on [Security Assessment](http://www.oemr.org/wiki/Security_Assessment) for EMRs (especially regarding HIPAA compliance) on the OEMR Organization wiki.

**This document is still just a work in progress; hopefully as members of the community begin securing their OpenEMR instances for the web, they will place things they learned here in order to help others.**

OpenEMR

* After installation/upgrade consider removing(or ensuring no access to) to the following scripts, which are not needed for general OpenEMR use:
  + acl\_setup.php
  + acl\_upgrade.php
  + sl\_convert.php
  + setup.php
  + sql\_upgrade.php
  + gacl/setup.php
  + ippf\_upgrade.php
  + entire contrib directory
    - (except for the contrib/icd9, contrib/icd10, contrib/snomed and contrib/rxnorm directories)
* After installing a patch consider removing (or ensuring no access to) the following script, which is not needed for general OpenEMR use:
  + sql\_patch.php
* Passwords
  + Enforce strong/unique passwords, which can be set in Administration->Globals->Security.
  + Enforce password expiration, which can be set in Administration->Globals->Security.

Network

* On server, consider only opening port 443 (https).
* Consider a firewall that only allows port 443 (https) traffic to the server.

Apache

* General hardening of Apache, which is [described here](http://www.learncomputer.com/secure-apache/).
* Only allow https (ie. turn off http)
* Do not allow direct web access to the following directories
  + sites/\*/documents
  + sites/\*/era
  + sites/\*/edi
* If not using portal and want to allow users to access over the internet, then consider using client-sided certificates to only allow users access to site
* If want to give access to the patient portals over the internet, then consider using a client-sided certificate to only allow users access to the main OpenEMR login page
* If using third party patient portal, then consider:
  + Only allowing the IP address of the third party portal to access the SOAP APIs (for patient access)
  + The third party portal could set up a certificate to identify their server as an option to further secure.
  + The third party portal could also set up a ssh tunnel method, so the local OpenEMR instance is not open to the web (and possibly not even require an IP address).

MySQL

* Ensure the root password is set to something.

PHP

* Follow the [OpenEMR php settings recommendations](http://www.open-emr.org/wiki/index.php/FAQ" \l "What_are_the_correct_PHP_settings_.28can_be_found_in_the_php.ini_file.29_.3F)

XAMPP

* If you are using XAMPP, note that this software is extremely unsecure after installation and you need to secure it (See the [XAMPP security instructions](http://www.apachefriends.org/en/xampp-windows.html#1221) to secure your XAMPP installation.).

Codebase

* [Ongoing project to prevent sql-injection and cross-scripting attacks.](http://www.open-emr.org/wiki/index.php/Codebase_Security)
* Consider adding a specific sqlconf.php override settings when using the patient portal (especially the onsite portal), which can be used to use a separate mysql user (and possibly database).

# Security Assessment

Now that the clinic is preparing to go electronic, the HIPAA Security Rule will apply. Compliance with these regulations and standard backup procedures will be essential.

### Risk management

Health information is an asset that needs to be protected. A risk assessment is the first step in continuing to protect this asset. An analysis should be made identifying the information assets, then determining the threats and vulnerabilities along with existing controls. Probabilities of the threats occurring in spite of the controls weighed against the impact of the occurrence will lead to an educated effort to protect the information asset.[[1]](http://www.oemr.org/wiki/Security_Assessment#cite_note-Herzig-0)

A security risk assessment is a requirement for compliance with HIPAA and Meaningful Use. [[2]](http://www.oemr.org/wiki/Security_Assessment#cite_note-HIMSS-1)[[3]](http://www.oemr.org/wiki/Security_Assessment#cite_note-2)[[4]](http://www.oemr.org/wiki/Security_Assessment#cite_note-3) The National Institute of Standards and Technology (NIST) outlines a nine step risk process: [[5]](http://www.oemr.org/wiki/Security_Assessment#cite_note-4)

* System characterization
* Threat identification
* Vulnerability identification
* Control analysis
* Likelihood determination
* Impact analysis
* Risk determination
* Control recommendations
* Results determination

The Healthcare Information and Management Systems Society (HIMSS) provides a [Risk Assessment Toolkit](http://www.himss.org/ASP/topics_PStoolkit_RiskAssessment.asp) for conducting a risk assessment and developing a compliant security plan. Available for download is the [Security Risk Assessment Guide/Data Collection Matrix](http://www.himss.org/ASP/ContentRedirector.asp?ContentID=79427) from the Risk Assessment Toolkit that can be tailored to the individual practice setting.

### Physical

Physical security protects the providers, the patients, clinic assets, and the physical manifestation of the business, financial, and health information. Traditional security matters involve locks, points of entry, lighting and cameras, fire precautions, power outage procedures, and water protection, and disaster recovery. Only after these measures have been taken can further policies in IT security be developed. The equipment storing PHI and BI should be subject to: Physical access restrictions: Place the server in a protected, locked area and limit access. Environmental precautions:

* Put equipment on a UPS to prevent data loss and downtime in a power outage, spike, or sag.
* Place server away from water sprinklers and out from under plumbing.
* Store physical backups on and off site in areas resistant to fire and water damage.
* Control static electricity around equipment using antistatic surfaces or sprays.
* Allow equipment to "breathe," placing away from walls or vents so heat may dissipate.

Anymore, not all of the equipment containing PHI remains within the walls of the clinic. Laptops, tablets, and smart phones all can bear PHI and these items are highly mobile. They should be assigned to responsible parties and loss or theft should be promptly reported and action to protect further security holes should be taken.

### Backup and disaster recovery

The organization should have a disaster recovery plan in order to provide continued patient care in the event of an unlikely and catastrophic event. A large part of this plan is a data backup strategy. Not only should the data be backed up, but it should be retrievable. A documented plan should include: [[1]](http://www.oemr.org/wiki/Security_Assessment#cite_note-Herzig-0)

* Backup frequency
* Backup type
* Media type
* Labeling convention
* Storage and transportation
* Rotation of media
* Retention
* Encryption
* Responsible parties
* Testing

Backup strategies not only should be compliant with HIPAA Security Rule specifications but with other regulations that oversee financial data (credit cards.)

### Access

Access controls ensure that users with a need to know the information in a system are able to access it, while unauthorized users cannot. A secure system has the ability to identify, then authenticate a user. In the most basic form, the identification is a username, and authentication is a password. The three basic authentication factors available for verification if identity are:

* Something the user *knows* (password or PIN)
* Something the user *has* (smart card or token)
* Something the user *is* (biometric)

Two-factor authentication uses two of these methods. For e-prescribing and for [CCHIT certification](https://www.cchit.org/c/document_library/get_file?uuid=3b3e0d39-b640-4ec0-b87b-2e2598059b1e&groupId=18), EHR systems shall have the ability to support two-factor authentication. Even in the most secure systems, there can be a weak link. Human factors are often the cause of access breaches, by some of these mechanisms:

* Weak passwords
  + Passwords should be held to a standard and be required to change at intervals. Weak policies allow users to set poor, breakable passwords. Stringent policies can frustrate users and lead to workarounds, like visible notes.
* Password sharing
  + While this is against security policy, users may feel this practice is necessary to deliver optimal patient care. Identify the workflow issue propagating this behavior.
* Phishing and social engineering
  + Phishing is a method of tricking users into entering or revealing credentials in an insecure location. Social engineering is a skillful manipulation of the user to reveal credentials. Staff must be educated on how to avoid these attacks.