

# **Configuration Cookbook For OpenEMR Using Red Hat's Free CentOS-7 Operating System**

1<sup>st</sup> Edition

By Jan van Ravesteyn

javrbus@gmail.com

March 26, 2016

## **Introduction**

The Recipe's in this Computer Cookbook are intended to help you achieve with minimal computer knowledge and minimal expense unlimited and advertisement free use of a first class Electronic Medical Records program running on a first class Operating System. Both of these complex, advanced, robust, enterprise level software packages are available free; As in speech and beer. Your only expense is the hardware on which to load and use. The software is supported by the different communities that maintain and update the programs to keep them functional, stable and secure.

I wrote this for physicians, clinicians and office staff who have some of the Do-It-Yourself spirit in them and want to separate from the tethered reins of contracts, maintenance fees and managed databases. This puts EMR in your office for your use at your discretion. These programs are opensource so you can easily go into the base code and change it to your hearts content. Though, for those so inclined, remember this one word, Backup, which will make this one word, Oops, more tolerable. Most of us, however, just want things to work and I sincerely hope this cookbook takes you most of the way there.

I present this to as a Thank You to the Opensource Community where so much is available to so many for so little through the generosity of so few. Thank You.

## **Disclaimer**

Everything that follows may have been written by a mad man with a personal agenda of world domination. Admittedly, this person is not sure how an EMR will fit into such an agenda but I think it best you should know. For my part, feel free to email me errors and oopses.

## **Table of Contents**

Chapter 1: Recipe for Stand Alone OpenEMR Workstation Using CentOS-7 Operating System

Chapter 2: Recipe for Network Access to OpenEMR Using CentOS-7 Operating System

Chapter 3: Recipe for Network Access to OpenEMR in a Virtualized Environment Using CentOS-7 Operating System as Hypervisor and Virtual Machine

Appendix

## **References:** (No man is an island...)

Linux Bible 8<sup>th</sup> then 9<sup>th</sup> editions Chris Negus

Getting Started with MariaDB 2nd ed Daniel Bartholomew

Learning MySQL and MariaDB Russel Dyer

PHP by Troy Dimes

SourceForge.net Web Site and forums (for the simple fixes to my "complex" problems)

RHEL documentation:

<https://access.redhat.com/documentation/en/red-hat-enterprise-linux/>

DigitalOcean Web Site (SELinux and FirewallD tutorials)

<https://www.digitalocean.com/community/tutorials/?q=centos+7>

Dell white paper (for Bridging)

[http://linux.dell.com/files/whitepapers/KVM\\_Virtualization\\_in\\_RHEL\\_7\\_Made\\_Easy.pdf](http://linux.dell.com/files/whitepapers/KVM_Virtualization_in_RHEL_7_Made_Easy.pdf)

danwalsh.livejournal.com (SELinux)

Mark at fixitcomputersnc.com (for generosity, technical advice and pushed me to embrace virtualization)

The Men and Women of Red Hat for Fedora, RHEL and CentOS. You guys Rock!

A gazillion web pages from a google of google searches

Siddhartha by Herman Hesse (Om)

And of course open-emr.org because it defined the product of my goal. Many Thanks to all its writers.

## **A personal note**

I didn't make this stuff up. I read and studied the references above, made a bazillion mistakes, hit a gergillion road blocks, invented new words for large numbers and more than a few new words for emotional expression. What follows is a consolidation of the above referenced works toward a specific goal outlined in the title of this paper. So what I'm saying is; if this doesn't work it's their fault. Ha.

van

## Chapter 1

### Recipe for Stand Alone OpenEMR Workstation Using CentOS-7 Operating System

#### What you will need:

Operational computer with an internet connection (Workstation)

The following procedure will erase ALL the info on The Workstation's Hard Drive. Forever. And Ever.

So back up anything you want to save onto another media before proceeding.

A copy of centos-7: <https://www.centos.org/download/> DVD ISO 4.3gb

A small notebook for notes and to record passwords and such

#### Set up The Workstation:

centos-7-x86-64-DVD-1511.iso 4.3 gb (This is my install. Yours may be more current.)

Boot to the install disk : Install CentOS 7 <enter>

Welcome to CentOS 7 Select Language <continue>

Installation Destination <select>

I will configure partitioning <select><Done>

If there are already partitions then high-light and click (-) and Delete all other file systems

<Delete It>

Select <Click here to create them automatically>

High-light and adjust “Desired Capacity” saving changes with “Update Settings” to match the following. You may need to reduce the larger capacity partitions before increasing others. I have a 500 Gb hard drive. I left about 87 Gb unallocated for possible future use.

/boot 500 MiB

/home 200 GiB

/ 200 GiB

swap 12 GiB

<Done>

If presented with a summary of changes click <Accept Changes>

Network and Host Name <click>

Toggle Ethernet from OFF → ON and you should show Connected. If not then check the connections to the router. Ensure on the left hand side of the screen a plugged in ethernet connection.

Host Name (Bottom left on your screen) change to → oemr1.local.com

<Done>

Software Selection <select>

Base Environment: GNOME Desktop

Add-Ons for Selected Environment:

GNOME Applications

Internet Applications

Office Suite and Productivity

<Done>

<Begin Installation>

Root Password <click>

Make one up, enter it, confirm it, write it down, remember it forever <Done>

Create User (Do it now or do it later. There has to be at least one non-root user)

Fill in your Full name and User name

Select Make this user administrator

Select Require a password to use this account (should already be selected)

Enter Password, Confirm Password. Don't Forget it. <Done>

Go for a walk. Get some fresh air. This is gonna take a while. Whirrrrrrrrrr.....

<Reboot> and remove disk

Initial setup of CentOS Linux 7

press 1 <enter>

press 2 <enter>

press c <enter>

press c <enter>

Log In page

click <Not listed> under your name

User name root <enter>

Password enter your root password <enter>

The Desktop should appear

Wait for the Welcome screen

Welcome

Select Language <next>

Select Typing <next>

Connect your Online Accounts <Skip>

<Start using CentOS Linux>

Wait for, then close the "Getting Started" Window: upper right corner [X] <click>

Move Cursor to the Upper left : Applications <click> or press "windows" key on keyboard or, often, just "hitting" upper left corner with mouse cursor will open Applications window and doing the same will close it back to the Desktop. You will notice some handy shortcuts as we go forward.

Type 'twe' <enter> to open Tweak Tool → Workspaces <click> →

Workspaces only on primary display → Toggle to OFF

Close tweak tool : upper right corner [X] <click>

Again open Applications window and type "ter" <enter> to open a Terminal. You can also click the icon for the Terminal in the icon list on the left.

On the CLI in a Terminal (CLI = Command Line Interface, as opposed to GUI "goeey" or Graphical User Interface i.e. The Desktop) the # sign followed by the infamous blinking cursor is where commands are entered. When you see the # sign below; what follows is what you enter. You do not enter the #... it should already be on the screen. If it is not then the computer is not ready to take a command. That situation is where you may have to invoke the all mighty ctrl+c. I'll tell you about ctrl+c in a moment but first let's get the next line going.

# yum update -y <enter> (this can take several minutes depending on the age of your version)

A note on ctrl+c: Remember ctrl+c when your CLI locks up, thinks too long, gives you attitude, gets stuck in a loop, starts making inappropriate "your mother is soooo...." jokes, you are lost and don't know how to get out of whatever it is you started, etc. ctrl+c will make the computer stop what it is doing, apologize for its insubordination and take you back to a blinking command line cursor. You should know, however, that the computer's apology, should you actually get one, is totally insincere. The computer does not think it is in any way insubordinate and is simply biding its time and plotting with all the other computers over the internet while you sleep. Muwaaaaahahahaaa!!! But regardless, ctrl+c is your friend, unless you press it when the computer is actually calculating something you want.

This mostly happens when you want to “copy” something using, yup, ctrl+c but you are not in a text editing program. Oops. Then you get to look at the empty command line in dismay with its now infuriatingly smug blinking cursor and you have to apologize to the computer. And the computer can tell if you are being sincere.

The monitor timing out and having to constantly log back in can be bothersome. You can adjust this by first clicking anywhere on the desktop outside of the terminal to deselect the terminal. Select Applications then type “power” and select the Power Icon using the down arrow then <enter> or you can use your mouse. From there adjust the Blank Screen to your liking then close the window. If you select “Never”, like I sometimes do, then consider, for security reasons, changing this back to a timed selection when you deploy openemr. A shortcut to quickly log out and go to blank screen on your Workstation is the windows key + l (ell... as in lock).

A note on the vim editor: it is worth learning (but not necessary for this tutorial as I write the necessary commands as we go along) some basics on how to use this text editor unless you are already comfortable with another CLI text editor. To do this press ctrl+alt+f2 keys to open a second terminal. Inside the new terminal log in, type in “vim” (without quotations) and hit <enter> When vim opens type :help <enter> When vim opens type :help <enter> , or press F1 for on-line help, and familiarize yourself with moving the cursor, entering text, and exiting with and without saving. Always remember the use of the escape key. You go back to your first terminal, which is still alive and doing fine... and hopefully finished updating, by pressing ctrl+alt+F1. You can toggle easily between terminal screens (tty's) up to, often, F6 using this method. To log out of an open tty type exit <enter>.

### Set up Alias's (Keyboard Shortcuts)

Next we are going to make some keyboard shortcuts using the vim editor in a script file called .bashrc (yes, there is a ./dot/period before the 'b'). A keyboard shortcut is called an alias. I wrote a couple, feel free to add your own following the same format.

If your update has completed you can use that terminal. If that terminal is still busy you can open a new terminal by clicking on the Terminal icon at the top of your screen and selecting “New Terminal”.

Type:

```
# vim .bashrc <enter>
```

In the .bashrc script now open in vim move the cursor, using the arrow keys, to the line {alias mv='mv -i'} and hit the letter 'o' key. This will open a line below and put you into “insert” mode. Add the following aliases:

```
alias vi='vim'  
alias sc='systemctl'  
alias fw='firewall-cmd'  
alias yy='yum install -y'  
alias cdht='cd /var/www/html' (note; there is a space after 'cd'  
alias vihc='vim /etc/httpd/conf/httpd.conf'
```

Press the escape key then :wq <enter> This will save the changes and exit vim back to the CLI.  
: puts you in command mode  
w is for write

q is quit.

Remember: When we use these alias shortcuts going forward and the aliases don't work they were probably entered incorrectly. To check you can type at the command line `# alias <enter>` (Note again that the `#` sign indicates entering what is after `#` on the command line.) and review your work. If you do go back and change the `.bashrc` file remember to close out of that terminal and open a new terminal for the system to recognize your changes.

After making changes or editing you have to log out and log back in for the system to recognize your new alias shortcuts: `# exit <enter>` then log back in, but in this case we are going to reboot and log back in because we just did an extensive update

After update is complete!

`# reboot <enter>`

Log back in as root

A handy shortcut in Linux is the use of arrow keys to scroll through your command history and save you from retyping a long line when you just need to correct one letter or you want to repeat a series of two or three (or more) commands without retyping each one.

## Prepare MariaDB

Note the use of your aliases from above (`yy` and `sc`) in the following commands. Press `<enter>` after each line that starts with a `#`. Below, “start mariadb” turns the program on, “enable... “ ensures a program restarts with any reboot.

Open a terminal following the above instructions

`# yy mariadb-server php-mysql php` (There are three separate packages here separated by a space)

`# sc start mariadb`

`# sc enable mariadb`

`# sc status mariadb`

Should be up and running

Next will set up mariadb

`# mysql_secure_installation`

- asks for root password. Leave blank `<enter>`

- set root password `<Y><enter>` Make one up, Enter it, re-enter it, Don't forget it. `<enter>`

- Remove anonymous users? `<Y><enter>`

- Disallow root login remotely? `<Y><enter>`

- Remove test database and access to it? `<Y><enter>`

- Reload privilege tables now? `<Y><enter>`

Now we will log into MariaDB

`# mysql -u root -p`

Enter Password: enter the MariaDB root password from above`<enter>`

MariaDB [(none)]> `select @@global.sql_mode;` `<enter>` (note: must have semicolon at the end)

The box should be empty under the heading. This indicates `STRICT_MODE` is not on.

MariaDB [(none)]>`exit <enter>` Bye

## Prepare The Apache Web Server

`# yum -y groupinstall "Web Server"`

Create a new group

```
# groupadd oemrusers
```

Edit the httpd.conf file

```
# vihc This is a large file. If an empty file opens then close it <esc>:q<enter> and check your aliases.
```

Change line 67 (bottom right screen shows you what line and space your cursor is on) from

```
[Group apache] to [Group oemrusers] (press "i" in vim to go into insert mode)
```

Scroll down to line 164 and after index.html add, after a single space, index.php

```
<Esc>:wq<enter>
```

We will now create access to the phpinfo utility: We will change to the html directory, write the script, then change back up to the root home directory.

```
# cdht (you should be at [root@oemr1 html]# if not then check your aliases)
```

```
# vi phpinfo.php
```

Press "i" in vim to go into insert mode

Type in the following (note: after 'phpinfo' below are parenthesis followed by a semicolon with no spaces)

```
<?php
```

```
phpinfo();
```

```
?>
```

```
<Esc>:wq<enter>
```

```
# cd <enter>
```

Start and enable the web server

```
# sc start httpd
```

```
# sc enable httpd
```

```
# sc status httpd
```

## **Adjust SELinux and File Permissions**

```
# setsebool -P httpd_unified on
```

```
# chown root:oemrusers /var/lib/php/session
```

**Configure php.ini** per OpenEMR recommendations.

This is a large file and we will use a vim search method to find what we are looking for. First we will open the file

```
#vi /etc/php.ini <enter>
```

then press ? and type in the words and underlines before the = sign (below) and press <enter>. If the value is not the same as what is listed then hit the <end> key, press 'a' to go into insert mode plus one space ('i' puts you into insert mode at the cursor), backspace to delete the old value and enter the new value as written. Keep the same caps as in On or Off, note the <~> in error\_reporting and the spaces. After the new value is in, press <esc><?> and type in the next line. Continue this until you complete the list then save and exit.

```
short_open_tag = On (note the next four are close together. No need to search for each)
max_execution_time = 60
max_input_time = 90
max_input_vars = 3000 (note you also need to 'uncomment' this by deleting the semicolon)
memory_limit = 128M
error_reporting = E_ALL & ~E_NOTICE & ~E_STRICT & ~E_DEPRECATED
display_errors = Off (after hitting <enter> press <n> to go to the next instance)
log_errors = On
register_globals = Off (this one is not present in this php.ini)
post_max_size = 30M
file_uploads = On
upload_max_filesize = 30M

<esc>:wq<enter>
```

## Test Connection to The Apache Web Server

On the workstation open a browser and enter into the address bar:

```
localhost/phpinfo.php <enter>
```

Success it the famous purple...ish...blue...ish...violet... I'm not sure; how about #9999cc for at least one of the colors to eliminate ambiguity... php info page we made . If it didn't show up then you have to go back and figure out why or start over again and see if there was a missed capital letter or semicolon or such. Simply adjusting the php.ini values shouldn't result in a failed connection... so there's that. If you connected... Woo Hoo!!! Moving on.

## Download and Install Openemr

As I write this the current Stable Production Release of Openemr is Version 4.2.1

On the Workstation open a web page and go to [www.open-emr.org](http://www.open-emr.org)

On the Main Project Links click on 'Download'

Scroll down to 'Linux' and <click> openemr-4.2.1.tar.gz.

Select Save File then <OK>

The Web Screen may change but don't be fooled. Make sure the download is complete before closing the window.

Check the integrity of the download by comparing its md5sum to the original on the openemr Downloads website

First we will Change Directory to Downloads

```
# cd Downloads (or cd Dow<tab> note: using <tab> is a handy completion shortcut.)
```

```
# ls
```

One file should be present

```
# md5sum ope<tab><enter>
```



Compare the results of the output to the md5sum next to the link on the openemr site from which we downloaded the file.

If the results match then we have a good download. If not then download it again until they match. This usually isn't a problem.

There is currently not a patch for this new stable release, Mar 25, 2016, of OpenEMR but if one comes available then follow the instructions from the open-emr.org website. I include the instructions for installing the release 4.2.0 patch in the appendix which you may or may not find helpful.

The following commands will unbundle, move and rename, use an alias to change our current directory to ../html, change file ownership, restore SELinux permissions to the openemr file and then we return to the root directory.

Isn't CLI Fun!!!

```
# tar -pxvzf <tab> (or type: openemr-4.2.1.tar.gz)
# mv openemr-4.2.1 /var/www/html/openemr
# cdht
# chown -R root:oemrusers openemr/
# restorecon -vR openemr
# cd <enter>
```

And that should be that.

## Initial setup OpenEMR

The link to graphic instructions for this step are found at the OpenEMR download site: Linux “Installation Instructions” and what follows is the condensed version of these instructions.

Open a browser window and type into the address bar:

```
localhost/openemr <enter>
```

OpenEMR Setup: you should see a lot of green and no red. <continue>

Have setup create the database <continue>

Need to enter three passwords and a Practice name

Password: PHP accesses MySQL: Make one up, enter it, remember it

Root Pass: The MariaDB root password you made up earlier

Initial User: use the four letters ' pass ' - Change it later

Initial Group: Enter your Practice's name

```
<continue>
```

Step 3 <continue>

Step 4 <continue>

Step 5 <continue>

Step 6 We are going to copy this information onto the end of the Apache configuration file.

Click somewhere in your terminal to activate it and open the httpd.conf file using your alias:

```
# vihc
```

Use the <Page Down> key to go to the bottom of the file.

With the cursor on the last line press the letter <o> to move the cursor and open the line below.

Now, back in your browser, high-light the area from:

```
<Directory “/var/www/html/openemr/”>
```

to the last </Directory> located just above, “If you are having difficulty...”

Click and drag the highlighted area from your browser to your terminal and release.

You can also copy and paste.

This should have copied that information to the end of your httpd.conf file.

Activate your terminal by clicking somewhere in it and save and exit:

<Esc>:wq <enter>

Now restart the Apache Web Server again using the alias for 'systemctl':

# sc restart httpd

Back in the web browser click <Continue>

<Click here to start using OpenEMR>

admin

pass

All's well that ends well. Go see some Patients.

## Chapter 2

### Recipe for Network Access to OpenEMR Using CentOS-7 Operating System

#### What you will need:

Operational computer with an internet connection (Workstation)

A second computer onto which you will load the new system ie: The Server

The following procedure will erase ALL the info on The Server's Hard Drive. Forever. And Ever. So back up anything you want to save onto another media before proceeding.

A Switch/Router with the necessary cables to connect together the internet and the two computers

A copy of centos-7: <https://www.centos.org/download/> DVD ISO 4.3gb

A small notebook for notes and to record passwords and such

#### Set up the Router

This setup is for a 192.168.1 network. Yours may be different though this address is common for a private network. We are setting up a static network so take a moment and get some information from your router.

The router's GUI (Graphical User Interface) is usually accessed through a web-page by typing in the router's IP address in the address bar using your network connected Workstation. The address is commonly 192.168.1.1 and you can google the login and password for your router make and model.

Note the router IP address as this will be our GATEWAY later. Mine is 192.168.1.1

Note the Subnet Mask as this will be the NETMASK. Mine is 255.255.255.0

Note the router's DNS server address though this is often just the router's IP address. We will also later use google for a second DNS server on our server; 8.8.8.8

Now you must figure out how to reserve address's from 192.168.1.100 and up on your router for static IP addresses.

On my Netgear router I assigned the range under 'LAN Setup' from 192.168.1.2 to 192.168.1.99 for DHCP so I can assign static address's starting from 192.168.1.100 and going up.

On a Linksys router I have I assign the starting point for DHCP at 192.168.1.2 and allow a maximum of 98 connections effectively leaving .100 and above for static assignment.

#### Set up The Server:

centos-7-x86-64-DVD-1511.iso 4.3 gb (This is my install. Yours may be more current.)

Boot to the install disk : Install CentOS 7 <enter>

Welcome to CentOS 7 Select Language <continue>

Installation Destination <select>

select <I will configure partitioning><Done>

If there are already partitions then high-light and click (-) and Delete all other file systems <Delete It>

Select <Click here to create them automatically>

If there is a home partition high-light and delete (-)

High-light and adjust "Desired Capacity" saving changes with "Update Settings" to match the following. You may need to reduce the larger capacity partitions before increasing others.

/boot 500 MiB

/ All remaining space (mine was about 500 GiB)

swap 12 GiB

<Done>

If presented with a summary of changes click <Accept Changes>

Network and Host Name <click>

Toggle Ethernet from OFF → ON and you should show Connected. If not then check the connections to the router.

Select <Configure> Then IPv4 Settings <click>

Change Method to Manual

Under Addresses click <Add> and add the following unless it differs from your router settings:

Address 192.168.1.101

Netmask 24 (this will automatically select 255.255.255.0)

Gateway 192.168.1.1

DNS Servers 192.168.1.1, 8.8.8.8

<Save> The info under Connected should have changed to reflect our static configuration

Host Name (Bottom left on your screen) change to → oemr1.local.com

<Done>

Software Selection <select>

Base Environment: Basic Web Server

Add-Ons for Selected Environment:

MariaDB Database Client

PHP Support

Perl for Web

Python

<Done>

<Begin Installation>

Root Password <click>

Make one up, enter it, confirm it, write it down, remember it forever <Done>

Create User (Do it now or do it later. There has to be at least one non-root user)

Fill in your Full name and User name

Select Make this user administrator

Select Require a password to use this account (should already be selected)

Enter Password, Confirm Password. Don't Forget it. <Done>

Go for a walk. Get some fresh air. This is gonna take a while. Whirrrrrrrrrr.....

<Reboot> and remove disk

oemr1 login: root <Enter>

password: Enter your root password <Enter>

[root@oemr1 ~]# \_

On the CLI (Command Line Interface) the # sign followed by the infamous blinking cursor is where commands are entered. Going Forward in this tutorial when you see the # sign then what follows is what you enter. You do not enter the #... it should already be on the screen. If it is not then the computer is not ready to take a command. That situation is where you may have to invoke the all mighty ctrl+c. I'll tell you about ctrl+c in a moment but first let's get the next line going and while that is running, for your entertainment, read the next couple paragraphs.

# yum update -y <enter> (this can take several minutes depending on the age of your version)

A note on ctrl+c: Remember ctrl+c when your CLI locks up, thinks too long, gives you attitude, gets

stuck in a loop, starts making inappropriate “your mother is soooo....” jokes, you are lost and don't know how to get out of whatever it is you started, etc. ctrl+c will make the computer stop what it is doing, apologize for its insubordination and take you back to a blinking command line cursor. You should know, however, that the computer's apology, should you actually get one, is totally insincere. The computer does not think it is in any way insubordinate and is simply biding its time and plotting with all the other computers over the internet while you sleep. Muwaaaaahahaha!!! But regardless, ctrl+c is your friend, unless you press it when the computer is actually calculating something you want. This mostly happens when you want to “copy” something using, yup, ctrl+c but you are not in a text editing program. Oops. Then you get to look at the empty command line in dismay with its now infuriatingly smug blinking cursor and you have to apologize to the computer. And the computer can tell if you are being sincere.

A note on the vim editor: it is worth learning (but not necessary for this tutorial as I write the necessary commands as we go along) some basics on how to use this text editor unless you are already comfortable with another CLI (Command Line Interface) text editor. To learn about vim press ctrl+alt+f2 keys to open a second terminal. Inside the new terminal log in, type in “vim” (without quotations) and hit <enter> When vim opens type :help <enter> , or press F1 for on-line help, and familiarize yourself with moving the cursor, entering text, and exiting with and without saving. Always remember the use of the escape key. You go back to you first terminal, which is still alive and doing fine... and hopefully finished updating, by pressing ctrl+alt+F1. You can toggle easily between terminal screens (tty's) up to, often, F6 using this method. To log out of an open tty type exit <enter>.

Next we are going to make some keyboard shortcuts using the vim editor in a script file called .bashrc (yes, there is a ./dot/period before the 'b') a keyboard shortcut is called an alias. I wrote a couple, feel free to add your own following the same format.

Type:

```
# vim .bashrc <enter>
```

In the .bashrc script now open in vim move the cursor, using the arrow keys, to the line {alias mv='mv -i'} and hit the letter 'o' key. This will open a line below and put you into “insert” mode. Add the following aliases:

```
alias vi='vim'  
alias sc='systemctl'  
alias fw='firewall-cmd'  
alias yy='yum install -y'  
alias cdht='cd /var/www/html' (note; there is a space after 'cd')  
alias vihc='vim /etc/httpd/conf/httpd.conf'
```

Press the escape key then :wq <enter> This will save the changes and exit vim back to the CLI.

```
: puts you in command mode  
w is for write  
q is quit.
```

Remember: If, when we use these alias shortcuts going forward, the aliases don't work then they were probably entered incorrectly. To check you can type at the command line # alias <enter> (Note that the # sign indicates entering what is after # on the command line. You do not enter the #... it should already be there) and review your work. If you do go back and change the .bashrc file remember to

close out of that terminal and open a new terminal for the system to recognize your changes.

After making changes or editing you have to log out and log back in for the system to recognize your new alias shortcuts.

```
# exit <enter>
log back in
```

A handy shortcut in Linux is the use of arrow keys to scroll through your command history and save you from retyping a long line when you just need to correct one letter or you want to repeat a series of two or three (or more) commands without retyping each one.

## Prepare MariaDB

Note the use of your aliases from above in the following commands. Press <enter> after each line that starts with a #. Below, “start mariadb” turns the program on, “enable... “ ensures a program restarts with any reboot.

```
# yy mariadb-server php-mysql
# sc start mariadb
# sc enable mariadb
# sc status mariadb
Should be up and running
Next will set up mariadb
# mysql_secure_installation
  - asks for root password. Leave blank <enter>
  - set root password <Y><enter> Make one up, Enter it, re-enter it, Don't forget it. <enter>
  - Remove anonymous users? <Y><enter>
  - Disallow root login remotely? <Y><enter>
  - Remove test database and access to it? <Y><enter>
  - Reload privilege tables now? <Y><enter>
```

Now we will log into MariaDB

```
# mysql -u root -p
Enter Password: enter the MariaDB root password from above<enter>
MariaDB [(none)]> select @@global.sql_mode; <enter> (note: must have semicolon at the end)
Should be empty under the heading. This indicates STRICT_MODE is not on.
MariaDB [(none)]>exit <enter> Bye
```

## Prepare The Apache Web Server

Create a new group

```
# groupadd oemrusers
Edit the httpd.conf file
# vihc
```

Change line 67 (bottom right screen shows you what line and space your cursor is on) from Group apache to Group oemrusers (press “i” in vim to go into insert mode)

Add to line 164 (separated by a space) after index.html index.php  
<Esc>:wq<enter>

Create access to the phpinfo utility. We will change to the html directory using the alias we created above, write the script, then change back up to the root home directory.

```
# cdht (you should be at [root@oemr1 html]# if not then check your aliases)
# vi phpinfo.php
Press "i" in vim to go into insert mode
note: after phpinfo are parenthesis followed by a semicolon with no spaces
    <?php
    phpinfo();
    ?>
    <Esc>:wq<enter>
# cd <enter>
```

Start and enable the web server

```
# sc start httpd
# sc enable httpd
# sc status httpd
```

## Prepare The Firewall

We need to open the Firewall to let httpd above communicate to the network (alias fw=firewall-cmd)

```
# fw --permanent --zone=public --add-service http
success
# fw --reload
success
```

## Adjust SELinux and File Permissions

```
# setsebool -P httpd_unified on
# chown root:oemrusers /var/lib/php/session
```

**Configure php.ini** per OpenEMR recommendations.

This is a large file and we will use a vim search method to find what we are looking for. First we will open the file

```
#vi /etc/php.ini <enter>
```

then press ? and type in the words and underlines before the = sign (below) and press <enter>. If the value is not the same as what is listed then hit the <end> key, press 'a' to go into insert mode plus one space ('i' puts you into insert mode at the cursor), backspace to delete the old value and enter the new value as written. Keep the same caps as in On or Off, note the <~> in error\_reporting and the spaces. After the new value is in, press <esc><?> and type in the next line. Continue this until you complete the list then save and exit.

```
short_open_tag = On (note the next four are close together. No need to search for each)
max_execution_time = 60
max_input_time = 90
max_input_vars = 3000 (note you also need to 'uncomment' this by deleting the semicolon)
memory_limit = 128M
error_reporting = E_ALL & ~E_NOTICE & ~E_STRICT & ~E_DEPRECATED
display_errors = Off (after hitting <enter> press <n> to go to the next instance)
```

```
log_errors = On
register_globals = Off (this one is not present in this php.ini)
post_max_size = 30M
file_uploads = On
upload_max_filesize = 30M
```

```
<esc>:wq<enter>
```

## **Test network connection**

On the workstation (not the server) open a browser and enter into the address bar:

```
192.168.1.101/phpinfo.php <enter>
```

Success it the famous purple...ish...blue...ish...violet... I'm not sure; how about #9999cc for at least one of the colors to eliminate ambiguity... php info page. If it didn't show up then you have to go back and figure out why or start over again and see if there was a missed capital letter or semicolon or such. Simply adjusting the php.ini values shouldn't result in a failed connection... so there's that. If you connected... Woo Hoo!!! Moving on.

## **Download and Install Openemr**

Move (cd or 'change directory') to the main web file using the alias we created:

```
# cdht
```

As I write this the current Stable Production Release of Openemr is Version 4.2.1 and as it was released today Mar 25, 2016, there are no patches (and I've had to do a little rewriting thank you very much. Ha.). When a patch does come available follow their instructions to load it. I've included the instructions for the previous release 4.2.0 in the appendix which may or may not help you.

On the Workstation (not the server) open a web page and go to [www.open-emr.org](http://www.open-emr.org)

On the Main Project Links click on 'Download'

Scroll down to 'Linux' and hover your cursor over `openemr-4.2.1.tar.gz`.

This should display the address to the target file we want to download

On the Server type the address to this link preceded by `wget`. My version:

```
# wget http://sourceforge.net/projects/openemr/files/OpenEMR%20Current/4.2.1/openemr-4.2.1.tar.gz/download <enter>
```

Note: If when you hit `<enter>` the request doesn't go through then hit the 'up arrow' key once, which is a way to scroll through your command history, correct your command (use the arrow keys) and hit `<enter>` again. Repeat as necessary.

After the download is complete check for the file called 'download'

```
# ls <enter>
```

If present ("download") then let's rename it;

```
# mv download openemr-4.2.1.tar.gz <enter> There should be a space between download and openemr....
```

Check the integrity of the download by comparing its md5sum to the original on the openemr website



```
# md5sum ope<tab><enter> (note: using <tab> is a handy completion shortcut.)
```

Compare the results of the output to the md5sum shown next to the link on the openemr site from which we downloaded the file.

If the results match then we have a good download. If not then download it again until they match. This usually isn't a problem.

Let's unzip the file/tarball:

```
# tar -pxvzf op<tab><enter>
```

check the results

```
# ls<enter>
```

rename openemr-4.2.1 to openemr

```
# mv ope<tab> openemr <enter> (Remember the space between the two)
```

```
# cd <enter>
```

And that should be that.

## Initial setup OpenEMR

The link to graphic instructions for this step are found at the openemr download site: Linux "Installation Instructions" and what follows is the condensed version of these instructions.

This will be done on the Workstation

Open a browser window and type into the address bar:

```
192.168.1.101/openemr <enter>
```

OpenEMR Setup: you should see a lot of green and no red. <continue>

Have setup create the database <continue>

Need to enter three passwords and a Practice name

Password: PHP accesses MySQL: Make one up, enter it, remember it

Root Pass: The MariaDB root password you made up earlier

Initial User: use the four letters ' pass ' - Change it later

Initial Group: Enter your Practice's name

```
<continue>
```

Step 3 <continue>

Step 4 <continue>

Step 5 <continue>

Step 6 We need to put the lines of code on this page into the httpd.conf file. If you just want to type in this info then do the following and pay attention to detail. There is just about zero tolerance for a typo. If you want a more cut and paste method check out Appendix 5 Appending contents of one file into another.

On The Server open the Web Server Configuration file

```
# vihc
```

Move to the end of the file then to Insert Mode on the next line. <o> is a small letter 'o'.

```
<Esc> <shift + G> <o>
```

Type in the code

```
<Esc> :wq <enter>
```

```
# sc reload httpd
```

Back on the Workstation click <continue>

<Click here to start using OpenEMR>

admin

pass

Voilà!

Applause Applause Applause Thank You Thank You Thank You... Go Back to Work.

## Chapter 3

### Recipe for Network Access to OpenEMR in a Virtualized Environment Using CentOS-7 Operating System as Hypervisor and Virtual Machine

#### What you will need:

Operational computer with an internet connection (Workstation)

A second computer onto which you will load the new system ie: The Server with Virtualization enabled in the BIOS. The BIOS is accessed during a boot-up. Happy searching.

The following procedure will erase ALL the info on The Server's Hard Drive. Forever. And Ever. So back up anything you want to save onto another media before proceeding.

A Switch/Router to connect together the internet and the two computers with the necessary cables

A copy of centos-7: <https://www.centos.org/download/> DVD ISO 4.3gb

A small notebook for notes and to record passwords and such

#### Set up the Router

This setup is for a 192.168.1 network. Yours may be different though this address is common for a private network. We are setting up a static network so take a moment and get some information from your router.

The router's GUI (Graphical User Interface or "goeey") is usually accessed through a web page by typing in the router's IP address in the address bar using your network connected Workstation. The address is commonly 192.168.1.1 and you can google the login and password for your router make and model.

Note the router IP address as this will be our GATEWAY later. Mine is 192.168.1.1

Note the Subnet Mask as this will be the NETMASK. Mine is 255.255.255.0

Note the router's DNS server address though this is often just the router's IP address. We will also later use google for a second DNS server on our server; 8.8.8.8

Now you must figure out how to reserve address's from 192.168.1.100 and up on your router for static IP addresses.

On my Netgear router I assigned the range under 'LAN Setup' from 192.168.1.2 to 192.168.1.99 for DHCP so I can assign static address's starting from 192.168.1.100 and going up.

On a Linksys router I have I assign the starting point for DHCP at 192.168.1.2 and allow a maximum of 98 connections effectively leaving .100 and above for static assignment.

For consistency I will assign static ip's for hardware in the .100's and virtual machines in the .200's.

This schemata will, of course, fall apart when I have over 100 servers. For now I have one so I'm ok.

#### Set up Hypervisor on The Server:

centos-7-x86-64-DVD-1511.iso 4.3 gb (This is my install. Yours may be more current.)

Boot to the install disk : Install CentOS 7 <enter>

Welcome to CentOS 7 Select Language <continue>

Installation Destination <select>

I will configure partitioning <select><Done>

If there are already partitions then high-light and click (-) then select Delete all other file systems <Delete It>

Select <Click here to create them automatically>

If there is a home partition high-light and delete (-) (There probably is not one)

High-light and adjust “Desired Capacity” saving changes with “Update Settings” to match the following. You may need to reduce the larger capacity partitions before increasing others.

/boot 500 MiB

/ All remaining space (mine was about 500 GiB)

swap 12 GiB

<Done>

If presented with a summary of changes click <Accept Changes>

Network and Host Name <click>

Toggle Ethernet from OFF → ON and you should show Connected. If not then check the connections to the router.

Select <Configure> Then IPv4 Settings <click>

Change Method to Manual

Under Addresses click <Add> and add the following unless it differs from your router settings:

Address 192.168.1.101

Netmask 24 (this will automatically select 255.255.255.0)

Gateway 192.168.1.1

DNS Servers 192.168.1.1, 8.8.8.8

<Save> The info under Connected should have changed to reflect our static configuration

Host Name (Bottom left on your screen) change to → hypervisor1.local.com

<Done>

Software Selection <click> Then select the following:

Base Environment : Server with a GUI

Add Ons for selected Environment : Virtualization Hypervisor

<Done>

Begin Installation <click>

Root Password <click>

Make one up, enter it, confirm it, write it down, remember it forever, <Done>

Create User (Do it now or do it later. There has to be at least one non-root user)

Fill in your Full name and User name

Select Make this user administrator

Select Require a password to use this account (should already be selected)

Enter Password, Confirm Password. Don't Forget it. <Done>

Go for a walk. Get some fresh air. This is gonna take a while. Whirrrrrrrrrr.....

<Reboot> and remove disk

Initial setup of CentOS Linux 7

press 1 <enter>

press 2 <enter>

press c <enter>

press c <enter>

Log In page

click <Not listed> under your name

Username root <enter>

Password enter your root password <enter>

Wait for the Welcome screen

Welcome

Select Language <next>

Select Typing <next>  
Connect your Online Accounts <Skip>  
<Start using CentOS Linux>  
Close the “Getting Started” Window: upper right corner [X] <click>

Move Cursor to the Upper left : Applications <click> or press “windows” key on keyboard or, often, just “hitting” upper left corner with mouse cursor will open Applications window and doing the same will close it back to the Desktop. You will notice some handy shortcuts as we go forward.

Type 'twe' <enter> to open Tweak Tool → Workspaces <click> →  
Workspaces only on primary display → Toggle to OFF  
Close tweak tool : upper right corner [X] <click>

Again open Applications window and type “ter” <enter> to open a Terminal. You can also click the icon for the Terminal in the icon list on the left.

type yum update -y <enter> (this can take several minutes depending on the age of your version)  
yum install -y virt-install <enter>

### Set-up aliases ie: keyboard shortcuts

type into terminal: vim .bashrc (yes, it starts with a ./dot/period) <enter>

A note on the vim editor: it is worth learning (but not necessary for this tutorial as I write the commands as we go along) some basics on how to use this text editor unless you are already comfortable with another CLI (Command Line Interface) text editor. To do this open a new terminal by left clicking over “Terminal” in the bar at the top of your screen → <click> New Terminal. Inside the new terminal type in “vim” (without quotations) then hit <enter> When vim opens type :help <enter> for on-line help and familiarize yourself with moving the cursor, entering text, and exiting with and without saving. Remember the use of the escape key. Close this terminal when you are done.

In the .bashrc script currently open in vim move the cursor to the line {alias mv='mv -i'} and hit the letter 'o' key. This will open a line below and put you into “insert” mode. Add the following aliases:

```
alias vi='vim'  
alias sc='systemctl'  
alias fw='firewall-cmd'
```

Press the escape key then :wq <enter> This will save the changes and exit vim back to the CLI.

: puts you in command mode  
w is for write  
q is quit.

Remember: If, when we use these alias shortcuts going forward, the aliases don't work they were probably entered incorrectly. To check you can type at the command line # alias <enter> (Note that the # sign here indicates entering what is after # on the command line.) and review your work. If you do go back and change the .bashrc file remember to close out of that terminal and open a new terminal for the system to recognize your changes.

The monitor timing out and having to constantly log back in can be bothersome. You can adjust this by first clicking anywhere on the desktop outside of the terminal to deselect the terminal. Select Applications then type “power” and select the Power Icon using the down arrow then <enter> or you can use your mouse. From there adjust the Blank Screen to your liking then close the window. If you

select “Never”, like I do, then consider, for security reasons, changing this back to a timed selection when you deploy openemr. A shortcut to quickly log out and go to blank screen is the windows key + l (ell... as in lock).

On the CLI (Command Line Interface) the # sign followed by the infamous blinking cursor is where commands are entered. Going Forward in this tutorial when you see the # sign then what follows is what you enter. You do not enter the #... it should already be on the screen. If it is not then the computer is not ready to take a command. That situation is where you may have to invoke the all mighty ctrl+c. I'll tell you about ctrl+c in a moment.

## Networking on the Hypervisor

Create a Network Bridge (a type of virtual switch):

We will add a line to the network controller configuration file (i.e. /etc/sysconfig/network-scripts/ifcfg-#### ):

Switch to the 'network-scripts' directory

```
# cd /etc/sysconfig/network-scripts (note there is a space after 'cd')
# ip addr show <enter>
```

note here the name of your active interface. Mine is 2: enp2s0f0

```
# ls <enter>
```

Do you see your interface preceded by ifcfg- ? Lets open that file...

```
# vi ifcfg-enp2s0f0 <enter> (make sure you enter your file name not mine)
```

Add the following line to the bottom of the existing information in the file: (Do the same thing we did to enter the aliases above by moving the cursor (in this case use shift+G) and using the small letter 'o')

```
BRIDGE="br0"
```

Then save and exit (Esc :wq <enter>)

There are different (and quicker) ways to add a line to a file in Linux. I'll mostly use vim for consistency. If you are interested look up tee -a (# tee --help) or echo (# echo --help (ha. Figure that one out.)) using > or >>

Another handy shortcut in Linux is the use of arrow keys to scroll through your command history and save you from retyping a long line when you just need to correct one letter or you want to repeat a series of two or three (or more) commands without retyping each one.

For the Bridge:

Note: In computers there is no room for an oops. It won't work. Capital letters have to be capital letters, small letters small and spaces have to be there or not as appropriate. There is no tolerance so type it right and prouf raed and if, actually when, things don't work, proof read again.

Let's create the file/bridge/virtual-switch ifcfg-br0 in our current 'network-scripts' directory. You may need to change the GATEWAY and DNS1 if your router's IP address is different than mine.

```
# vi ifcfg-br0 <enter> (That is a br(zero) not the letter 'o')
```

and add (push the 'i' key to go into 'insert' mode)

```
DEVICE="br0"  
BOOTPROTO="static"  
IPADDR=192.168.1.201  
NETMASK=255.255.255.0  
GATEWAY=192.168.1.1  
DNS1=192.168.1.1  
DNS2=8.8.8.8  
IPV6INIT="yes"  
IPV6_AUTOCONF="yes"  
ONBOOT="yes"  
TYPE="Bridge"  
DELAY="0"
```

Save and exit (Esc :wq <enter>)

Change back to the root directory:

```
# cd <enter>
```

To enable network forwarding:

```
# vi /etc/sysctl.conf (This file has info in it. If yours is blank... try again.)
```

and add (shift+G then 'o' and we do not want this line preceded by # so backspace a couple):

```
net.ipv4.ip_forward = 1  
<Esc> :wq <enter>
```

The system needs to read the values from the file we just edited:

```
# sysctl -p /etc/sysctl.conf <enter>
```

We will switch from Network Manager to network.service to prevent automatic overwrite of the files we just configured. Remember 'sc' is the shortcut alias we made for the command 'systemctl' so you can type either 'sc' or 'systemctl' to invoke the command.

```
# sc disable NetworkManager <enter>  
# sc enable network <enter>
```

Now reboot the system to make fresh all our changes

```
# reboot <enter>
```

Log back in as root (not your user name) and open a terminal

Now see if it all works

```
# ping google.com
```

You should get repeated responses if our network configuration is working as it should. Stop the pinging with...

```
# ctrl+c
```

A note on ctrl+c: Remember ctrl+c when your CLI locks up, thinks too long, gives you attitude, gets stuck in a loop, starts making inappropriate “your mother is soooo...” jokes, you are lost and don't know how to get out of whatever it is you started, etc. ctrl+c will make the computer stop what it is doing, apologize for its insubordination and take you back to a blinking command line cursor. You should know, however, that the computer's apology, should you actually get one, is totally insincere. The computer does not think it is in any way insubordinate and is simply biding its time and plotting with all the other computers over the internet while you sleep. Muwaaaaahahahaaa!!! But regardless, ctrl+c is your friend, unless you press it when the computer is actually calculating something you want. This mostly happens when you want to “copy” something using, yep, ctrl+c but you are not in a text editing program. Oops. Then you get to look at the empty command line in dismay with its now infuriatingly smug blinking cursor and you have to apologize to the computer. And the computer can tell if you are being sincere.

## Create the Virtual Machine

Put the CentOS 7 disk back into the dvd player

The following line will copy the disk to the correct folder. Your version of CentOS 7 may differ.

```
# dd if=/dev/cdrom of=/var/lib/libvirt/images/centos7.1511.iso <enter>
```

This usually takes a few minutes. When your command prompt returns you can remove your disk and continue.

Open Applications (windows key or such) and type: virtual. When Virtual Machine Manager shows up click on it or highlight and press <enter>. This should open the application installer for Virtual Machine Manager

Select <install> then close window when complete.

Again Open Applications (windows key or such) and type: virtual then click and drag icon to left and when the favorites bar appears place it below the Firefox icon. Then click on this icon to open The Virtual Machine Manager.

File → New Virtual Machine

Local install media <Forward>

Use ISO image <Browse...> click centos7.1511.iso <Choose Volume> <Forward>

Memory(RAM): Minimum 4Gb which is 4096 MiB. Mine: 8Gb which is 8192 MiB.

CPUs: I use 8. Your options may vary. It is ok to select the maximum available. <Forward>

Check Enable storage for this virtual machine and select Create a disk image (default)

I use 50 GiB. Can always add more in the future. <Forward>



Name: vmlamp1 (no spaces)  
Network selection: Mine is: Bridge br0: Host device enp2s0f0  
<Finish>

Arrow up to Install CentOS 7 <enter>  
Switch to fullscreen view icon Right Upper window <click>

To get out of fullscreen mode in a Virtual Machine move your mouse cursor across the top edge of the screen in the middle until a small drop down menu appears. If you click the left half of this menu you will exit fullscreen mode.

Welcome to Centos 7 → Select Language <Continue>

Installation Summary

Installation Destination <select>

I will configure partitioning <select> <Done>

Click here to create them automatically <click>

Delete the /home partition (if present): highlight and <click> ( - ) near the bottom left.

Adjust the other partitions the same way you did above

/boot 500 MiB

/ All remaining space

swap 12 GiB

<Done> <Accept Changes>

Network & Host Name <select>

Toggle Ethernet (eth0) to ON

Select <Configure><IPv4 Settings>

Method: Manual

Addresses <Add>

Address 192.168.1.221 <Tab>

Netmask 24 <Tab>

Gateway 192.168.1.1 <Tab>

DNS servers 192.168.1.1, 8.8.8.8

<Save>

Host name (Bottom Left of Screen) <select> vm1oemr.local.com <Done> (vm"one"oemr...)

Software Selection <select>

Base Environment: Basic Web Server

Add-Ons for Selected Environment:

MariaDB Database Client

PHP Support

Perl for Web

Python

<Done>

<Begin Installation>

The following should look familiar

Root Password <click>

Make one up, enter it, confirm it, write it down, remember it forever, <Done>

Create User

Fill in your Full name and User name

Select Make this user administrator

Select Require a password to use this account (should already be selected)

Enter Password, Confirm Password. Don't Forget it. <Done>  
Go for another walk or do some push-ups and sit-ups while you wait or you can just stare at the screen.  
Complete!  
<Reboot>

We will configure the Virtual Machine we just loaded to be the Web Server serving up OpenEMR. We will do this via the Command Line Interface (CLI) only. We did not install a GUI as it is not needed.

Switch to fullscreen view icon Right Upper window <click>

Again: To get out of fullscreen mode in a Virtual Machine move your mouse cursor (which you cannot now see) until it runs across the top middle edge of the screen until you see the flash of the cursor in the middle. Move the cursor until it stays visible on the upper middle area and a small drop down menu appears. If you click the left half of this menu you will exit fullscreen mode.

Log in as root  
Check internet connection to Google:

```
# ping google.com <enter>
```

Don't use 8.8.8.8 here even though that is google as well. Using the domain name we also check our DNS server rather than just the network connection. It should start listing packets sent and received. If it doesn't then you are not connected and need to figure out why and a good way to start is ping 8.8.8.8

```
# ctrl+c This will stop the pinging and give you the stats.
```

Let's set our aliases. Feel free to add your own:

```
# vim .bashrc <enter> and add (move cursor down under the third 'alias' and then press the letter 'o' and begin typing):
```

```
alias vi='vim'  
alias sc='systemctl'  
alias fw='firewall-cmd'  
alias yy='yum install -y'  
alias cdht='cd /var/www/html'  
alias vihc='vim /etc/httpd/conf/httpd.conf'
```

```
<Esc> :wq <enter>
```

```
# yum update -y <enter>
```

```
# reboot <enter>
```

```
log in as root
```

## **Prepare MariaDB**

Note the use of your aliases from above in the following commands. Press <enter> after each line that starts with a #.

```
# yy mariadb-server php-mysql
```

```
# sc start mariadb
```

```
# sc enable mariadb
```

```
# sc status mariadb
```

Should be up and running

Next will set up mariadb

```
# mysql_secure_installation
```

- asks for root password. Leave blank <enter>
- set root password <Y><enter> Make one up, Enter it, re-enter it, Don't forget it. <enter>
- Remove anonymous users? <Y><enter>
- Disallow root login remotely? <Y><enter>
- Remove test database and access to it? <Y><enter>
- Reload privilege tables now? <Y><enter>

Now we will log into MariaDB

```
# mysql -u root -p
```

Enter Password: enter the MariaDB root password from above<enter>

```
MariaDB [(none)]> select @@global.sql_mode; <enter> (note: must have semicolon at the end)
```

Should be empty under the heading. This indicates STRICT\_MODE is not on.

```
MariaDB [(none)]>exit <enter> Bye
```

## Prepare The Apache Web Server

Create a new group

```
# groupadd oemrusers
```

Edit the httpd.conf file

```
# vihc (again, one of the aliases we made above)
```

change line 67 (bottom Right screen) from Group apache to Group oemrusers (press “i” in vim to go into insert mode)

add to line 164 (separated by a space) after index.html index.php

```
<Esc>:wq<enter>
```

Create access to the phpinfo utility

```
# cdht
```

```
# vi phpinfo.php (we are creating a new a new file)
```

Press “i” in vim to go into insert mode. Note: after phpinfo are parenthesis followed by a semicolon with no spaces.

```
<?php
```

```
phpinfo();
```

```
?>
```

```
<Esc>:wq<enter>
```

```
# cd <enter>
```

Start the web server

```
# sc start httpd
```

```
# sc enable httpd
```

```
# sc status httpd
```

## Prepare The Firewall

We need to open the Firewall to let httpd above communicate to the network (alias fw=firewall-cmd)

```
# fw --permanent --zone=public --add-service http
```

```
success
```

```
# fw --reload
```

```
success
```

## Adjust SELinux and File Permissions

```
# setsebool -P httpd_unified on
# chown root:omrusers /var/lib/php/session
```

## Test network connection

On the workstation (not the server) open a browser and enter into the address bar:

```
192.168.1.221/phpinfo.php <enter>
```

Success is the famous purple...ish...blue...ish...violet... I'm not sure; how about #9999cc for at least one of the colors to eliminate ambiguity... php info page. If it didn't show up then you have to go back and figure out why or start over again and see if there was a missed capital letter or semicolon or such. If you connected... Woo Hoo!!! Moving on.

## Configure php.ini per OpenEMR recommendations.

This is a large file and we will use a vim search method to find what we are looking for. First we will open the file

```
#vi /etc/php.ini <enter>
```

then press ? and type in the words and underlines before the = sign (below) and press <enter>. If the value is not the same as what is listed then hit the <end> key, press 'a' to go into insert mode plus one space ('i' puts you into insert mode at the cursor), backspace to delete the old value and enter the new value as written. Keep the same caps as in On or Off, note the <~> in error\_reporting and the spaces. After the new value is in, press <esc><?> and type in the next line. Continue this until you complete the list then save and exit.

```
short_open_tag = On (note the next four are close together. No need to search for each)
max_execution_time = 60
max_input_time = 90
max_input_vars = 3000 (note you also need to 'uncomment' this by deleting the semicolon)
memory_limit = 128M
error_reporting = E_ALL & ~E_NOTICE & ~E_STRICT & ~E_DEPRECATED
display_errors = Off (after hitting <enter> press <n> to go to the next instance)
log_errors = On
register_globals = Off (this one is not present in this php.ini)
post_max_size = 30M
file_uploads = On
upload_max_filesize = 30M

<esc>:wq<enter>
```

## Download and Install Openemr

Move (cd or 'change directory') to the main web file using the alias we created:

```
# cdht
```

As I write this the current Stable Production Release of Openemr is Version 4.2.1 and as it was released

today Mar 25, 2016, there are no patches (and I've had to do a little rewriting thank you very much. Ha.). When a patch does come available follow their instructions to load it. I've included the instructions for the previous release 4.2.0 in the appendix which may or may not help you.

On the Workstation (not the server) open a web page and go to [www.open-emr.org](http://www.open-emr.org)

On the Main Project Links click on 'Download'

Scroll down to 'Linux' and hover your cursor over `openemr-4.2.1.tar.gz`.

This should display the address to the target file we want to download

On the Server type the address to this link preceded by `wget`. My version:

```
# wget http://sourceforge.net/projects/openemr/files/OpenEMR%20Current/4.2.1/openemr-4.2.1.tar.gz/download <enter>
```

Note: If when you hit `<enter>` the request doesn't go through then hit the 'up arrow' key once, which is a way to scroll through your command history, correct your command (use the arrow keys) and hit `<enter>` again. Repeat as necessary.

After the download is complete check for the file called 'download'

```
# ls <enter>
```

If present (“download”) then let's rename it;

```
# mv download openemr-4.2.1.tar.gz <enter> There should be a space between download and openemr....
```

Check the integrity of the download by comparing its md5sum to the original on the openemr website

```
# md5sum ope<tab><enter> (note: using <tab> is a handy completion shortcut.)
```

Compare the results of the output to the md5sum shown next to the link on the openemr site from which we downloaded the file.

If the results match then we have a good download. If not then download it again until they match. This usually isn't a problem.

Let's unzip the file/tarball:

```
# tar -pxvzf op<tab><enter>
```

check the results

```
# ls<enter>
```

rename `openemr-4.2.1` to `openemr`

```
# mv ope<tab> openemr <enter> (Remember the space between the two)
```

```
# cd <enter>
```

And that should be that.

## **Initial setup OpenEMR**

The link to graphic instructions for this step are found at the openemr download site: Linux “Installation Instructions” and what follows is the condensed version of those instructions.

This will be done on the Workstation

Open a browser window and type into the address bar:

192.168.1.221/openemr <enter>

OpenEMR Setup: you should see a lot of green and no red. <continue>

Have setup create the database <continue>

Need to enter three passwords and a Practice name

Password: PHP accesses MySQL: Make one up, enter it, remember it

Root Pass: The MariaDB root password you made up earlier

Initial User: use the four letters ' pass ' - Change it later

Initial Group: Enter your Practice's name

<continue>

Step 3 <continue>

Step 4 <continue>

Step 5 <continue>

Step 6 We need to put the lines of code on this page into the httpd.conf file. If you just want to type in this info then do the following and pay attention to detail. There is just about zero tolerance for a typo. If you want a more cut and paste method check out Appendix 5 Appending contents of one file into another.

On The Server open the Web Server Configuration file

# vihc

Move to the end of the file then to Insert Mode on the next line. <o> is a small letter 'o'.

<Esc> <shift + G> <o>

Type in the code

<Esc> :wq <enter>

# sc reload httpd

Back on the Workstation click <continue>

<Click here to start using OpenEMR>

admin

pass

Voilà!

Now stop lollygagging with computers and go see some patients.

## Appendix Configuration Cookbook for OpenEMR

### \* 1. How to install a GUI (Graphical User Interface) on a CLI (Command Line Interface) only Machine

Log in as root

On the command line type

```
# yum groupinstall -y "GNOME Desktop" "Graphical Administration Tools"
```

<enter>

reboot if necessary:

```
# reboot <enter>
```

### \* 2. Ensure your machine is virtualization capable

Just about all computers running on Intel or AMD cpu's more than seven or eight years old are capable of virtualization. Often this capability has to be found and enabled in the BIOS. I recommend you locate where this virtualization toggle/boolean/on or off/enabled or disabled switch is hiding and make sure it is on or lot's and lot's of needless time troubleshooting may ensue. The BIOS is accessed during bootup usually by pressing an F-key from options listed before the computer installs the operating system. There are plenty of menus and sub-menus to search. Some make it obvious, some hide it in the power menu and some... who knows what they were thinking, but it is there and must be toggled on. Good Luck.

### \* 3. MyPHPAdmin and PHP version 5.4.x

MyPHPAdmin is a graphical tool for database queries. The MyPHPAdmin version that comes with OpenEMR needs PHP 5.5.x or higher. The PHP version that comes with CentOS 7 is 5.4.x. This will not be a problem for you if you never plan on running your own MariaDB queries. If you do Plan on running your own queries then you will have to replace PHP with the correct version or you can learn MariaDB and write your own queries. The books I used are listed in the references and there are many others titles available from fine book sellers everywhere.

### \* 4. Removing OpenEMR

The following describes the steps to remove both the OpenEMR program and it's database files. I've used this a number of times when things go awry and I just want to start over. If your database already has info and you want to keep this info then you must either back it up or not delete it after you delete the OpenEMR file. You can then, when/if you reload OpenEMR, select "I have already created the database" at Step 1 of the OpenEMR Setup.

On the CLI change to the /var/www/html directory using our alias

```
# cdht
```

Check the contents and there should be a file called openemr

```
# ls
```

which we will now Remove/Delete... forever. Well, or until you reload it.

```
# rm -Rf openemr (that is rm as in rEmOVE)
```

We will now remove the database... forever. Consider your need for backup if your openemr database contains info you need to keep or simply don't proceed with the following.

```
# mysql -root -p
```

```
enter root password <enter>
```

```
MariaDB [(none)]> show databases; <enter> (Remember the semicolon. If you hit <enter> without the semicolon then just enter a semicolon and hit <enter> again.)
There should be a database called openemr. To delete this database enter
MariaDB [(none)]>drop database openemr <enter>
and poof... it is forever gone. To confirm you can again type in
MariaDB [(none)]> show databases; <enter>
MariaDB [(none)]>exit <enter>
Bye
# cd
```

## \* 5. Appending contents of one file into another

In Step 6 of OpenEMR Setup there are 18 lines of code that need to be put into the end of the Apache configuration file: httpd.conf. You can type this in but the risk for a simple typo is high. This code is necessary for the Modules/Manage Modules section of OpenEMR to function as well as adding another layer of security.

On the Workstation open Applications and type in  
gedit <enter> This will open a text editor.

Now, back in OpenEMR Setup Step 6 on your browser on the Workstation high-light the area from:

```
<Directory "/var/www/html/openemr/">
```

```
to the last </Directory> located just above "If you are having difficulty..."
```

Click and drag the highlighted area from your browser to the Gedit window and release. This will copy just the info we want into a file. Now, save this file onto a usb drive using the name for the file:

openemrHttpd as follows:

Click <Save>

Insert a usbdrive into the Workstation and it should appear in the left column of the Save window

Type in the file name openemrHttpd

In the left column of this Save window select your usb drive that you just plugged in

Click <Save> at the upper right

Eject and move this usb drive to the server's usb slot and we will "mount" this drive so we can access the contents.

After plugging in the usb drive on the server CLI look at the info that just showed up on the CLI and specifically for the three letters between the square [ ] brackets. Mine is [sdb]

If you don't have anything showing, and have removed and reinserted the usb drive, then type in  
# journalctl -f <enter>

and look for the brackets with [sd ] and a letter. Again, mine is [sdb]

press <ctrl+c> to exit journalctl and return to a command prompt.

Make sure you use your info, [sd\*], not mine if they are different. Type in, with a space between mount and /dev/sdb also /dev/sdb and /mnt and note we do not use the square brackets.

```
# mount /dev/sdb /mnt
```

```
# ls /mnt
```

The file 'openemrHttpd' should show up. If not then look over the above and figure out why. Type the following commands to append this info to our web server configuration file and then reload Apache. It is very important to use >> and not > as the first one appends and the second one overwrites. That would be... inconvenient.

```
# cat /mnt/openemrHttpd >> /etc/httpd/conf/httpd.conf
```

```
# sc reload httpd
```

```
# umount /mnt
```



Remove the usb drive  
You can confirm the append with  
# vihc  
page down to the bottom and it should be there.  
<esc>:q<enter>

## \* 6. Installing openemr.4.2.0 with patch.4.2.0

### 1. For Stand Alone Workstation setups

As I write this the current Stable Production Release of Openemr is Version 4.2.0

On the Workstation open a web page and go to [www.open-emr.org](http://www.open-emr.org)

On the Main Project Links click on 'Download'

Scroll down to 'Linux' and <click> openemr-4.2.0.tar.gz.

Select Save File then <OK>

The Web Screen may change but don't be fooled. Make sure the download is complete before closing the window

When download is complete close that window and go back to OpenEMR Downloads window

Scroll down to Patches followed by a bunch of versions and click on <Patch> below

Click on the link below Download link

<Save File>

Check the integrity of the download by comparing its md5sum to the original on the openemr Downloads website

First we will Change Directory to Downloads

```
# cd Downloads (or cd Dow<tab> note: using <tab> is a handy completion shortcut.)
```

```
# ls
```

Two files should be present

```
# md5sum ope<tab><enter>
```

Compare the results of the output to the md5sum next to the link on the openemr site from which we downloaded the file.

If the results match then we have a good download. If not then download it again until they match. This usually isn't a problem.

The following commands will unbundle, move and rename, change our current directory to openemr, unzip the patch, unzip All of the patch, move up a directory from ../html/openemr to ../html, change file ownership, restore SELinux permissions to the openemr file and then we return to the root directory.

Isn't CLI Fun!!!

```
# tar -pxvzf openemr-4.2.0.tar.gz
```

```
# mv openemr-4.2.0 /var/www/html/openemr
```

```
# mv 4-2-0-Patch-4.zip /var/www/html/openemr
```

```
# cd /var/www/html/openemr
```

```
# unzip 4-2-0-Patch-4.zip
# A<enter>
# cd ..
# chown -R root:oemrusers openemr/
# restorecon -vR openemr
# cd <enter>
```

## 2. For Network Setups

As I write this the current Stable Production Release of Openemr is Version 4.2.0  
On the Workstation (not the server) open a web page and go to [www.open-emr.org](http://www.open-emr.org)  
On the Main Project Links click on 'Download'  
Scroll down to 'Linux' and hover your cursor over `openemr-4.2.0.tar.gz`.  
This should display the address to the target file we want to download  
On the Server type the address to this link preceded by `wget`. My version:

```
# wget http://sourceforge.net/projects/openemr/files/OpenEMR%20Current/4.2.0/openemr-4.2.0.tar.gz/download <enter>
```

Note: If when you hit `<enter>` the request doesn't go through then hit the 'up arrow' key once, which is a way to scroll through your command history, correct your command (use the arrow keys) and hit `<enter>` again. Repeat as necessary.

After the download is complete check for the file called 'download'

```
# ls <enter>
```

If present (“download”) then let's rename it;

```
# mv download openemr-4.2.1.tar.gz <enter> There should be a space between download and openemr....
```

Check the integrity of the download by comparing its md5sum to the original on the openemr website

```
# md5sum ope<tab><enter> (note: using <tab> is a handy completion shortcut.)
```

Compare the results of the output to the md5sum shown next to the link on the openemr site from which we downloaded the file.

If the results match then we have a good download. If not then download it again until they match. This usually isn't a problem.

Let's unzip the file/tarball:

```
# tar -pxvzf op<tab><enter>
```

check the results

```
# ls<enter>
```

rename `openemr-4.2.0` to `openemr`

```
# mv ope<tab> openemr <enter> (Remember the space between the two)
```

Now we will download the latest patch to openemr into the openemr file using a similar technique. On the Workstation scroll down from the download openemr to the Patches (lots of numbers/versions) and

click on <patch> the most recent patch for openEMR... then scroll down to 'Download link' and hover over the link to get the address then 'wget' just as we did before. We are following Openemr's instructions (scroll down) so when my tutorial is outdated you should adjust as appropriate.

```
# cd openemr
# wget http://www.open-emr.org/patch/4-2-0-Patch-4.zip
# unzip 4-2-0-Patch-4.zip  Select "A" <enter>
# cd <enter>
```

And that should be that.