Visual Midwifery

Support System Configuration

Prepared by

**Otago MeddIT**

Otago Polytechnic

Dunedin, New Zealand

Monday, 13 October 2014

Contents

[Visual Midwifery Support System Configuration 3](#_Toc400980623)

[Installation 3](#_Toc400980624)

[Host 3](#_Toc400980625)

[Sudoers 4](#_Toc400980626)

[Networking 4](#_Toc400980627)

[Software Installation**[Edit section](https://project.ict.op.ac.nz/Project/2014_Projects/February_Start/Otago_MeddIT/Iteration_Three/Support_System?action=edit&sectionId=5)** 5](#_Toc400980628)

[MySQL**[Edit section](https://project.ict.op.ac.nz/Project/2014_Projects/February_Start/Otago_MeddIT/Iteration_Three/Support_System?action=edit&sectionId=6)** 5](#_Toc400980629)

[SSL **[Edit section](https://project.ict.op.ac.nz/Project/2014_Projects/February_Start/Otago_MeddIT/Iteration_Three/Support_System?action=edit&sectionId=8)** 19](#_Toc400980630)

[Moodle 20](#_Toc400980631)

[SilverStripe 20](#_Toc400980632)

# Visual Midwifery Support System Configuration

## Installation

**Operating System:** Debian Linux 7.5 (Wheezy)

**Resource Allocation:**

**[Edit section](https://project.ict.op.ac.nz/Project/2014_Projects/February_Start/Otago_MeddIT/Iteration_Three/Support_System?action=edit&sectionId=2)**

**CPUs:** 1

**Memory:** 4GB

**Hard Disk:** 30GB

**System Configuration**

**[Edit section](https://project.ict.op.ac.nz/Project/2014_Projects/February_Start/Otago_MeddIT/Iteration_Three/Support_System?action=edit&sectionId=3)**

**Hostname:** meddit

**Domain:** ict.op.ac.nz

**Root password:** M3dL@mP

Default User:

**User:** steeljm1 (System Administration)

**Password:** XXX

**Admin User:**

**User Name:** puppetmaster

**Password:** 8907HD5=j%x7qwp9c3Dq

**Partitioning:**

sda       swap      8GB

                                  /           22GB

**Initial Software Selection:**

* SSH Server
* Standard system utilities

**Post Install Configuration**

**Host**

Set host name and domain:

# Edit /etc/hosts

|  |
| --- |
| # Set hostname and domain names  127.0.0.1       localhost       <%= hostname %>  127.0.1.1meddit.ict.op.ac.nzmeddit  #  Production static ip  10.25.1.160meddit.ict.op.ac.nz meddit    # The following lines are desirable for IPv6 capable hosts  ::1     localhost ip6-localhost ip6-loopback  ff02::1 ip6-allnodes  ff02::2 ip6-allrouters |

### Sudoers

Grant root access to the system administrator:

# Edit /etc/sudoers

|  |
| --- |
| # Please consider adding local content in /etc/sudoers.d/ instead of  # directly modifying this file.  #  # See the man page for details on how to write a sudoers file.  #  Defaultsenv\_reset  Defaultsmail\_badpass  Defaultssecure\_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"    # Host alias specification    # User alias specification    # Cmnd alias specification    # User privilege specification - Add system administrator  rootALL=(ALL:ALL) ALL  # Admin accounts  steeljm1 ALL=(ALL:ALL) ALL  puppetmaster ALL=(ALL:ALL) ALL    # Allow members of group sudo to execute any command  %sudoALL=(ALL:ALL) ALL    # See sudoers(5) for more information on "#include" directives:    #includedir /etc/sudoers.d |

### Networking

Set static ip address:

# Edit /etc/network/interfaces

|  |
| --- |
| # This file describes the network interfaces available on your system # and how to activate them. For more information, see interfaces(5).  # The loopback network interface auto lo iface lo inet loopback  # The primary network interface allow-hotplug eth0    # Apply appropriate addressing here iface eth0 inet static         address 10.25.1.160         netmask 255.255.0.0         gateway 10.25.0.3 |

## Software Installation[Edit section](https://project.ict.op.ac.nz/Project/2014_Projects/February_Start/Otago_MeddIT/Iteration_Three/Support_System?action=edit&sectionId=5)

### MySQL[Edit section](https://project.ict.op.ac.nz/Project/2014_Projects/February_Start/Otago_MeddIT/Iteration_Three/Support_System?action=edit&sectionId=6)

Install MySQL:

|  |
| --- |
| # apt-get install mysql-server mysql-client |

Create root password for mysql:

|  |
| --- |
| # /usr/bin/mysqladmin -u root password ${root\_password} |

# Edit /etc/mysql/debian-start

|  |
| --- |
| #!/bin/bash  #  # This script is executed by "/etc/init.d/mysql" on every (re)start.  #  # Changes to this file will be preserved when updating the Debian package.  #    source /usr/share/mysql/debian-start.inc.sh    MYSQL="/usr/bin/mysql --defaults-file=/etc/mysql/debian.cnf"  MYADMIN="/usr/bin/mysqladmin --defaults-file=/etc/mysql/debian.cnf"  MYUPGRADE="/usr/bin/mysql\_upgrade --defaults-extra-file=/etc/mysql/debian.cnf"  MYCHECK="/usr/bin/mysqlcheck --defaults-file=/etc/mysql/debian.cnf"  MYCHECK\_SUBJECT="WARNING: mysqlcheck has found corrupt tables"  MYCHECK\_PARAMS="--all-databases --fast --silent"  MYCHECK\_RCPT="root"    # The following commands should be run when the server is up but in background  # where they do not block the server start and in one shell instance so that  # they run sequentially. They are supposed not to echo anything to stdout.  # If you want to disable the check for crashed tables comment  # "check\_for\_crashed\_tables" out.  # (There may be no output to stdout inside the background process!)  echo "Checking for tables which need an upgrade, are corrupt or were "  echo "not closed cleanly."  (    upgrade\_system\_tables\_if\_necessary;    check\_root\_accounts;    check\_for\_crashed\_tables;  ) >&2 &    exit 0 |

# Eidt /etc/mysql/my.cf

|  |
| --- |
| #  # The MySQL database server configuration file.  #  # You can copy this to one of:  # - "/etc/mysql/my.cnf" to set global options,  # - "~/.my.cnf" to set user-specific options.  #  # One can use all long options that the program supports.  # Run program with --help to get a list of available options and with  # --print-defaults to see which it would actually understand and use.  #  # For explanations see  # [http://dev.mysql.com/doc/mysql/en/se...variables.html](http://dev.mysql.com/doc/mysql/en/server-system-variables.html)    # This will be passed to all mysql clients  # It has been reported that passwords should be enclosed with ticks/quotes  # escpecially if they contain "#" chars...  # Remember to edit /etc/mysql/debian.cnf when changing the socket location.  [client]  port= 3306  socket= /var/run/mysqld/mysqld.sock    # Here is entries for some specific programs  # The following values assume you have at least 32M ram    # This was formally known as [safe\_mysqld]. Both versions are currently parsed.  [mysqld\_safe]  socket= /var/run/mysqld/mysqld.sock  nice= 0    [mysqld]  #  # \* Basic Settings  #  user= mysql  pid-file= /var/run/mysqld/mysqld.pid  socket= /var/run/mysqld/mysqld.sock  port= 3306  basedir= /usr  datadir= /var/lib/mysql  tmpdir= /tmp  lc-messages-dir= /usr/share/mysql  skip-external-locking  #  # Instead of skip-networking the default is now to listen only on  # localhost which is more compatible and is not less secure.  ALLOW OTHER NETWORKS  bind-address= \*  #  # \* Fine Tuning  #  key\_buffer= 16M  max\_allowed\_packet= 16M  thread\_stack= 192K  thread\_cache\_size       = 8  # This replaces the startup script and checks MyISAM tables if needed  # the first time they are touched  myisam-recover         = BACKUP  #max\_connections        = 100  #table\_cache            = 64  #thread\_concurrency     = 10  #  # \* Query Cache Configuration  #  query\_cache\_limit= 1M  query\_cache\_size        = 16M  #  # \* Logging and Replication  #  # Both location gets rotated by the cronjob.  # Be aware that this log type is a performance killer.  # As of 5.1 you can enable the log at runtime!  #general\_log\_file        = /var/log/mysql/mysql.log  #general\_log             = 1  #  # Error logging goes to syslog due to /etc/mysql/conf.d/mysqld\_safe\_syslog.cnf.  #  # Here you can see queries with especially long duration  #log\_slow\_queries= /var/log/mysql/mysql-slow.log  #long\_query\_time = 2  #log-queries-not-using-indexes  #  # The following can be used as easy to replay backup logs or for replication.  # note: if you are setting up a replication slave, see README.Debian about  #       other settings you may need to change.  #server-id= 1  #log\_bin= /var/log/mysql/mysql-bin.log  expire\_logs\_days= 10  max\_binlog\_size         = 100M  #binlog\_do\_db= include\_database\_name  #binlog\_ignore\_db= include\_database\_name  #  # \* InnoDB  #  # InnoDB is enabled by default with a 10MB datafile in /var/lib/mysql/.  # Read the manual for more InnoDB related options. There are many!  #  # \* Security Features  #  # Read the manual, too, if you want chroot!  # chroot = /var/lib/mysql/  #  # For generating SSL certificates I recommend the OpenSSL GUI "tinyca".  #  # ssl-ca=/etc/mysql/cacert.pem  # ssl-cert=/etc/mysql/server-cert.pem  # ssl-key=/etc/mysql/server-key.pem        [mysqldump]  quick  quote-names  max\_allowed\_packet= 16M    [mysql]  #no-auto-rehash# faster start of mysql but no tab completition    [isamchk]  key\_buffer= 16M    #  # \* IMPORTANT: Additional settings that can override those from this file!  #   The files must end with '.cnf', otherwise they'll be ignored.  #  !includedir /etc/mysql/conf.d/ |

**Apache2**

**[Edit section](https://project.ict.op.ac.nz/Project/2014_Projects/February_Start/Otago_MeddIT/Iteration_Three/Support_System?action=edit&sectionId=7)**

Install Apache2 webserver:

|  |
| --- |
| # apt-get install apache2 apache2-doc |

Install other required packages:

|  |
| --- |
| # apt-get install php5 php5-gd php-xml-parser php5-intl php5-ldap libapache2-mod-php5 php-apc php5-mcrypt php5-mysql curl libcurl3 php5-curl php-soap php-xml-rpc openssl |

Enable required modules:

|  |
| --- |
| # a2enmod rewrite  # a2enmod headers  # a2enmod ssl |

# Edit /etc/apache2/apache2.conf

|  |
| --- |
| # This is the main Apache server configuration file.  It contains the  # configuration directives that give the server its instructions.  # See <http://httpd.apache.org/docs/2.2/> for detailed information about  # the directives and /usr/share/doc/apache2-common/README.Debian.gz about  # Debian specific hints.  #  #  # Summary of how the Apache 2 configuration works in Debian:  # The Apache 2 web server configuration in Debian is quite different to  # upstream's suggested way to configure the web server. This is because Debian's  # default Apache2 installation attempts to make adding and removing modules,  # virtual hosts, and extra configuration directives as flexible as possible, in  # order to make automating the changes and administering the server as easy as  # possible.    # It is split into several files forming the configuration hierarchy outlined  # below, all located in the /etc/apache2/ directory:  #  #/etc/apache2/  #|-- apache2.conf  #|`--  ports.conf  #|-- mods-enabled  #||-- \*.load  #|`-- \*.conf  #|-- conf.d  #|`-- \*  # `-- sites-enabled  # `-- \*  #  #  # \* apache2.conf is the main configuration file (this file). It puts the pieces  #   together by including all remaining configuration files when starting up the  #   web server.  #  #   In order to avoid conflicts with backup files, the Include directive is  #   adapted to ignore files that:  #   - do not begin with a letter or number  #   - contain a character that is neither letter nor number nor \_-:.  #   - contain .dpkg  #  #   Yet we strongly suggest that all configuration files either end with a  #   .conf or .load suffix in the file name. The next Debian release will  #   ignore files not ending with .conf (or .load for mods-enabled).  #  # \* ports.conf is always included from the main configuration file. It is  #   supposed to determine listening ports for incoming connections, and which  #   of these ports are used for name based virtual hosts.  #  # \* Configuration files in the mods-enabled/ and sites-enabled/ directories  #   contain particular configuration snippets which manage modules or virtual  #   host configurations, respectively.  #  #   They are activated by symlinking available configuration files from their  #   respective \*-available/ counterparts. These should be managed by using our  #   helpers a2enmod/a2dismod, a2ensite/a2dissite. See  #   their respective man pages for detailed information.  #  # \* Configuration files in the conf.d directory are either provided by other  #   packages or may be added by the local administrator. Local additions  #   should start with local- or end with .local.conf to avoid name clashes. All  #   files in conf.d are considered (excluding the exceptions noted above) by  #   the Apache 2 web server.  #  # \* The binary is called apache2. Due to the use of environment variables, in  #   the default configuration, apache2 needs to be started/stopped with  #   /etc/init.d/apache2 or apache2ctl. Calling /usr/bin/apache2 directly will not  #   work with the default configuration.      # Global configuration  #    #  # ServerRoot: The top of the directory tree under which the server's  # configuration, error, and log files are kept.  #  # NOTE!  If you intend to place this on an NFS (or otherwise network)  # mounted filesystem then please read the LockFile documentation (available  # at <URL:[http://httpd.apache.org/docs/2.2/mod....html#lockfile](http://httpd.apache.org/docs/2.2/mod/mpm_common.html#lockfile)>);  # you will save yourself a lot of trouble.  #  # Do NOT add a slash at the end of the directory path.  #  #ServerRoot "/etc/apache2"    #  # The accept serialization lock file MUST BE STORED ON A LOCAL DISK.  #  LockFile ${APACHE\_LOCK\_DIR}/accept.lock    #  # PidFile: The file in which the server should record its process  # identification number when it starts.  # This needs to be set in /etc/apache2/envvars  #  PidFile ${APACHE\_PID\_FILE}    #  # Timeout: The number of seconds before receives and sends time out.  #  Timeout 300    #  # KeepAlive: Whether or not to allow persistent connections (more than  # one request per connection). Set to "Off" to deactivate.  #  KeepAlive On    #  # MaxKeepAliveRequests: The maximum number of requests to allow  # during a persistent connection. Set to 0 to allow an unlimited amount.  # We recommend you leave this number high, for maximum performance.  #  MaxKeepAliveRequests 100    #  # KeepAliveTimeout: Number of seconds to wait for the next request from the  # same client on the same connection.  #  KeepAliveTimeout 5    ##  ## Server-Pool Size Regulation (MPM specific)  ##    # prefork MPM  # StartServers: number of server processes to start  # MinSpareServers: minimum number of server processes which are kept spare  # MaxSpareServers: maximum number of server processes which are kept spare  # MaxClients: maximum number of server processes allowed to start  # MaxRequestsPerChild: maximum number of requests a server process serves  <IfModule mpm\_prefork\_module>      StartServers          5      MinSpareServers       5      MaxSpareServers      10      MaxClients          150      MaxRequestsPerChild   0  </IfModule>    # worker MPM  # StartServers: initial number of server processes to start  # MinSpareThreads: minimum number of worker threads which are kept spare  # MaxSpareThreads: maximum number of worker threads which are kept spare  # ThreadLimit: ThreadsPerChild can be changed to this maximum value during a  #              graceful restart. ThreadLimit can only be changed by stopping  #              and starting Apache.  # ThreadsPerChild: constant number of worker threads in each server process  # MaxClients: maximum number of simultaneous client connections  # MaxRequestsPerChild: maximum number of requests a server process serves  <IfModule mpm\_worker\_module>      StartServers          2      MinSpareThreads      25      MaxSpareThreads      75      ThreadLimit          64      ThreadsPerChild      25      MaxClients          150      MaxRequestsPerChild   0  </IfModule>    # event MPM  # StartServers: initial number of server processes to start  # MinSpareThreads: minimum number of worker threads which are kept spare  # MaxSpareThreads: maximum number of worker threads which are kept spare  # ThreadsPerChild: constant number of worker threads in each server process  # MaxClients: maximum number of simultaneous client connections  # MaxRequestsPerChild: maximum number of requests a server process serves  <IfModule mpm\_event\_module>      StartServers          2      MinSpareThreads      25      MaxSpareThreads      75      ThreadLimit          64      ThreadsPerChild      25      MaxClients          150      MaxRequestsPerChild   0  </IfModule>    # These need to be set in /etc/apache2/envvars  User ${APACHE\_RUN\_USER}  Group ${APACHE\_RUN\_GROUP}    #  # AccessFileName: The name of the file to look for in each directory  # for additional configuration directives.  See also the AllowOverride  # directive.  #    AccessFileName .htaccess    #  # The following lines prevent .htaccess and .htpasswd files from being  # viewed by Web clients.  #  <Files ~ "^\.ht">      Order allow,deny      Deny from all      Satisfy all  </Files>    #  # DefaultType is the default MIME type the server will use for a document  # if it cannot otherwise determine one, such as from filename extensions.  # If your server contains mostly text or HTML documents, "text/plain" is  # a good value.  If most of your content is binary, such as applications  # or images, you may want to use "application/octet-stream" instead to  # keep browsers from trying to display binary files as though they are  # text.  #  # It is also possible to omit any default MIME type and let the  # client's browser guess an appropriate action instead. Typically the  # browser will decide based on the file's extension then. In cases  # where no good assumption can be made, letting the default MIME type  # unset is suggested  instead of forcing the browser to accept  # incorrect  metadata.  #  DefaultType None      #  # HostnameLookups: Log the names of clients or just their IP addresses  # e.g., www.apache.org (on) or 204.62.129.132 (off).  # The default is off because it'd be overall better for the net if people  # had to knowingly turn this feature on, since enabling it means that  # each client request will result in AT LEAST one lookup request to the  # nameserver.  #  HostnameLookups Off    # ErrorLog: The location of the error log file.  # If you do not specify an ErrorLog directive within a <VirtualHost>  # container, error messages relating to that virtual host will be  # logged here.  If you \*do\* define an error logfile for a <VirtualHost>  # container, that host's errors will be logged there and not here.  #  ErrorLog ${APACHE\_LOG\_DIR}/error.log    #  # LogLevel: Control the number of messages logged to the error\_log.  # Possible values include: debug, info, notice, warn, error, crit,  # alert, emerg.  #  LogLevel warn    # Include module configuration:  Include mods-enabled/\*.load  Include mods-enabled/\*.conf    # Include list of ports to listen on and which to use for name based vhosts  Include ports.conf    #  # The following directives define some format nicknames for use with  # a CustomLog directive (see below).  # If you are behind a reverse proxy, you might want to change %h into %{X-Forwarded-For}i  #  LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" vhost\_combined  LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combined  LogFormat "%h %l %u %t \"%r\" %>s %O" common  LogFormat "%{Referer}i -> %U" referer  LogFormat "%{User-agent}i" agent    # Include of directories ignores editors' and dpkg's backup files,  # see the comments above for details.    # Include generic snippets of statements  Include conf.d/    # Include the virtual host configurations:  Include sites-enabled/    # Include phpmyadmin  include /etc/phpmyadmin/apache.conf |

# Edit /etc/apache2/mods-available/php5.conf

|  |
| --- |
| # This file is managed by puppet    <FilesMatch ".+\.ph(p[345]?|t|tml)$">      SetHandler application/x-httpd-php  </FilesMatch>  <FilesMatch ".+\.phps$">      SetHandler application/x-httpd-php-source      # Deny access to raw php sources by default      # To re-enable it's recommended to enable access to the files      # only in specific virtual host or directory      Order Deny,Allow      Deny from all  </FilesMatch>  # Deny access to files without filename (e.g. '.php')  <FilesMatch "^\.ph(p[345]?|t|tml|ps)$">      Order Deny,Allow      Deny from all  </FilesMatch>    # Running PHP scripts in user directories is disabled by default  #  #To re-enable PHP in user directories comment the following lines  #(from <IfModule ...> to </IfModule>.) Do NOT set it to On as it  # prevents .htaccess files from disabling it.  <IfModule mod\_userdir.c>      <Directory /home/\*/public\_html>          php\_admin\_value engine Off      </Directory>  </IfModule> |

# Edit /etc/apache2/sites-enabled/000-default

|  |
| --- |
| NameVirtualHost \*:443  NameVirtualHost \*:80    <VirtualHost \*:80>  ServerAdmin webmaster@localhost  ServerName meddit.ict.op.ac.nz  DocumentRoot /var/www  <Directory />  Options FollowSymLinks  AllowOverride None  </Directory>  <Directory /var/www/>  Options Indexes FollowSymLinks MultiViews  AllowOverride None  Order allow,deny  allow from all  </Directory>    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/  <Directory "/usr/lib/cgi-bin">  AllowOverride None  Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch  Order allow,deny  Allow from all  </Directory>    ErrorLog ${APACHE\_LOG\_DIR}/error.log    # Possible values include: debug, info, notice, warn, error, crit,  # alert, emerg.  LogLevel warn    CustomLog ${APACHE\_LOG\_DIR}/access.log combined  </VirtualHost>      <VirtualHost \*:443>          ServerName meddit.ict.op.ac.nz            DocumentRoot /var/www/          ErrorLog /var/log/apache2/error.log          CustomLog /var/log/apache2/access.log combined            SSLEngine on          SSLCertificateFile /etc/ssl/localcerts/apache-meddit.pem  SSLCertificateKeyFile /etc/ssl/localcerts/apache-meddit.key  </VirtualHost> |

#Edit /etc/apache2/sites-enabled/default-ssl

|  |
| --- |
| <IfModule mod\_ssl.c>  <VirtualHost \_default\_:443>  ServerAdmin webmaster@localhost    DocumentRoot /var/www  <Directory />  Options FollowSymLinks  AllowOverride None  </Directory>  <Directory /var/www/>  Options Indexes FollowSymLinks MultiViews  AllowOverride None  Order allow,deny  allow from all  </Directory>    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/  <Directory "/usr/lib/cgi-bin">  AllowOverride None  Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch  Order allow,deny  Allow from all  </Directory>    ErrorLog ${APACHE\_LOG\_DIR}/error.log    # Possible values include: debug, info, notice, warn, error, crit,  # alert, emerg.  LogLevel warn    CustomLog ${APACHE\_LOG\_DIR}/ssl\_access.log combined    #   SSL Engine Switch:  #   Enable/Disable SSL for this virtual host.  SSLEngine on    #   A self-signed (snakeoil) certificate can be created by installing  #   the ssl-cert package. See  #   /usr/share/doc/apache2.2-common/README.Debian.gz for more info.  #   If both key and certificate are stored in the same file, only the  #   SSLCertificateFile directive is needed.  SSLCertificateFile    /etc/ssl/certs/ssl-cert-snakeoil.pem  SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key    #   Server Certificate Chain:  #   Point SSLCertificateChainFile at a file containing the  #   concatenation of PEM encoded CA certificates which form the  #   certificate chain for the server certificate. Alternatively  #   the referenced file can be the same as SSLCertificateFile  #   when the CA certificates are directly appended to the server  #   certificate for convinience.  #SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt    #   Certificate Authority (CA):  #   Set the CA certificate verification path where to find CA  #   certificates for client authentication or alternatively one  #   huge file containing all of them (file must be PEM encoded)  #   Note: Inside SSLCACertificatePath you need hash symlinks  #         to point to the certificate files. Use the provided  #         Makefile to update the hash symlinks after changes.  #SSLCACertificatePath /etc/ssl/certs/  #SSLCACertificateFile /etc/apache2/ssl.crt/ca-bundle.crt    #   Certificate Revocation Lists (CRL):  #   Set the CA revocation path where to find CA CRLs for client  #   authentication or alternatively one huge file containing all  #   of them (file must be PEM encoded)  #   Note: Inside SSLCARevocationPath you need hash symlinks  #         to point to the certificate files. Use the provided  #         Makefile to update the hash symlinks after changes.  #SSLCARevocationPath /etc/apache2/ssl.crl/  #SSLCARevocationFile /etc/apache2/ssl.crl/ca-bundle.crl    #   Client Authentication (Type):  #   Client certificate verification type and depth.  Types are  #   none, optional, require and optional\_no\_ca.  Depth is a  #   number which specifies how deeply to verify the certificate  #   issuer chain before deciding the certificate is not valid.  #SSLVerifyClient require  #SSLVerifyDepth  10    #   Access Control:  #   With SSLRequire you can do per-directory access control based  #   on arbitrary complex boolean expressions containing server  #   variable checks and other lookup directives.  The syntax is a  #   mixture between C and Perl.  See the mod\_ssl documentation  #   for more details.  #<Location />  #SSLRequire (    %{SSL\_CIPHER} !~ m/^(EXP|NULL)/ \  #            and %{SSL\_CLIENT\_S\_DN\_O} eq "Snake Oil, Ltd." \  #            and %{SSL\_CLIENT\_S\_DN\_OU} in {"Staff", "CA", "Dev"} \  #            and %{TIME\_WDAY} >= 1 and %{TIME\_WDAY} <= 5 \  #            and %{TIME\_HOUR} >= 8 and %{TIME\_HOUR} <= 20       ) \  #           or %{REMOTE\_ADDR} =~ m/^192\.76\.162\.[0-9]+$/  #</Location>    #   SSL Engine Options:  #   Set various options for the SSL engine.  #   o FakeBasicAuth:  #     Translate the client X.509 into a Basic Authorisation.  This means that  #     the standard Auth/DBMAuth methods can be used for access control.  The  #     user name is the `one line' version of the client's X.509 certificate.  #     Note that no password is obtained from the user. Every entry in the user  #     file needs this password: `xxj31ZMTZzkVA'.  #   o ExportCertData:  #     This exports two additional environment variables: SSL\_CLIENT\_CERT and  #     SSL\_SERVER\_CERT. These contain the PEM-encoded certificates of the  #     server (always existing) and the client (only existing when client  #     authentication is used). This can be used to import the certificates  #     into CGI scripts.  #   o StdEnvVars:  #     This exports the standard SSL/TLS related `SSL\_\*' environment variables.  #     Per default this exportation is switched off for performance reasons,  #     because the extraction step is an expensive operation and is usually  #     useless for serving static content. So one usually enables the  #     exportation for CGI and SSI requests only.  #   o StrictRequire:  #     This denies access when "SSLRequireSSL" or "SSLRequire" applied even  #     under a "Satisfy any" situation, i.e. when it applies access is denied  #     and no other module can change it.  #   o OptRenegotiate:  #     This enables optimized SSL connection renegotiation handling when SSL  #     directives are used in per-directory context.  #SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire  <FilesMatch "\.(cgi|shtml|phtml|php)$">  SSLOptions +StdEnvVars  </FilesMatch>  <Directory /usr/lib/cgi-bin>  SSLOptions +StdEnvVars  </Directory>    #   SSL Protocol Adjustments:  #   The safe and default but still SSL/TLS standard compliant shutdown  #   approach is that mod\_ssl sends the close notify alert but doesn't wait for  #   the close notify alert from client. When you need a different shutdown  #   approach you can use one of the following variables:  #   o ssl-unclean-shutdown:  #     This forces an unclean shutdown when the connection is closed, i.e. no  #     SSL close notify alert is send or allowed to received.  This violates  #     the SSL/TLS standard but is needed for some brain-dead browsers. Use  #     this when you receive I/O errors because of the standard approach where  #     mod\_ssl sends the close notify alert.  #   o ssl-accurate-shutdown:  #     This forces an accurate shutdown when the connection is closed, i.e. a  #     SSL close notify alert is send and mod\_ssl waits for the close notify  #     alert of the client. This is 100% SSL/TLS standard compliant, but in  #     practice often causes hanging connections with brain-dead browsers. Use  #     this only for browsers where you know that their SSL implementation  #     works correctly.  #   Notice: Most problems of broken clients are also related to the HTTP  #   keep-alive facility, so you usually additionally want to disable  #   keep-alive for those clients, too. Use variable "nokeepalive" for this.  #   Similarly, one has to force some clients to use HTTP/1.0 to workaround  #   their broken HTTP/1.1 implementation. Use variables "downgrade-1.0" and  #   "force-response-1.0" for this.  BrowserMatch "MSIE [2-6]" \  nokeepalive ssl-unclean-shutdown \  downgrade-1.0 force-response-1.0  # MSIE 7 and newer should be able to use keepalive  BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown    </VirtualHost>  </IfModule> |

### SSL [Edit section](https://project.ict.op.ac.nz/Project/2014_Projects/February_Start/Otago_MeddIT/Iteration_Three/Support_System?action=edit&sectionId=8)

# Create directory for ssl certificates

|  |
| --- |
| # mkdir -p /etc/ssl/localcerts |

**Self Signed certificates**

For a production deployment it will be necessary to have signed ssl certificates. These certificates need to named and placed into the appropriate directory. See the “Gernerate certificates“ section below for details.

Gernerate certificates:

|  |
| --- |
| # openssl req -new -x509 -days 365 -nodes -out **/etc/ssl/localcerts/apache-meddit.pem -keyout** **/etc/ssl/localcerts/apache-meddit.key** |

### 

### Moodle

Visual Midwifery is running version 2.6.2+ (Build: 20140320). It is recommended that you do upgrade to a newer release before reading the release notes for that particular version. We recommend ONLY upgrading to address security updates. If an upgrade is required ensure to preserve the configuration details in Moodle’s site configuration file:

|  |
| --- |
| /var/www/moodle/moodle/config.php |

It is also important to backup Moodle’s database in MySQL (DB name: moodle)and moodledata directory:

|  |
| --- |
| /var/moodledata |

The backup script provided will dump the database and archive the moodledata directory. See visualMidwiferyBackup-ver.1.2.sh for details.

Additional installation notes can be found on the official Moodle installation web page accessed here: <https://docs.moodle.org/23/en/Installing_Moodle>

### SilverStripe

Reinstalling SilverStripe should not be required; however, it is recommended that SilverStripe is installed using composer if necessary. Please follow these instructions to install SilverStripe: <http://doc.silverstripe.org/framework/en/installation/composer>

**Note:**

Before installing the new version ensure that the SilverStripe database, assets, and mysite directories are backed up.

Install Silverstripe into the root directory:

|  |
| --- |
| /var/www/ss |

Ensure that the settings in Visual Midwifery’s config file match the new installation:

|  |
| --- |
| /var/www/ss/mysite/\_config.php |

**Note:** Ensure that the **assets** directory is writable by www-data group

|  |
| --- |
| # chmod 775 /var/www/ss/assets |

|  |
| --- |
| # chown –R www-data:www-data /var/www/ss/assets |

The Visual Midwifery CMS requires the **“colourpicker”** plugin from Packagist: <https://packagist.org/packages/tractorcow/silverstripe-colorpicker>

To install colourpicker:

|  |
| --- |
| # cd /var/www/ss |

Run the following command:

|  |
| --- |
| # composer require: "tractorcow/silverstripe-colorpicker": "dev-master" |

After installing the plugin it is necessary to do a dev build:

|  |
| --- |
| http://meddit.ict.op.ac.nz/ss/index.php/dev/build |

The SilverStripe RESTful api is also required for serving stored data to the mobile app. The library and further information can be found here:

<https://github.com/silverstripe/silverstripe-restfulserver>