

This Guide is intended to help users install and configure Open Meetings 2.X

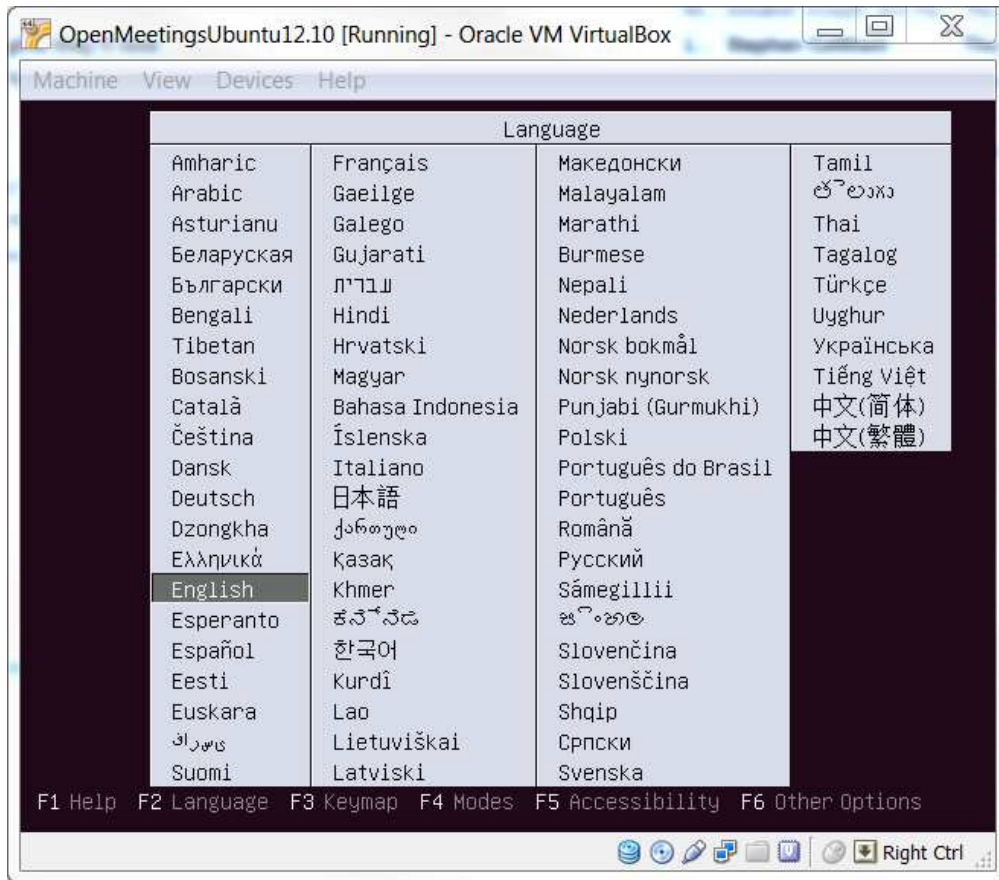
The guide is aimed at Ubuntu 12.10 users, it has been written step by step with screenshots to aid in the successful build of OM from a fresh install.

SSL and Reverse proxy steps have been added but are optional.

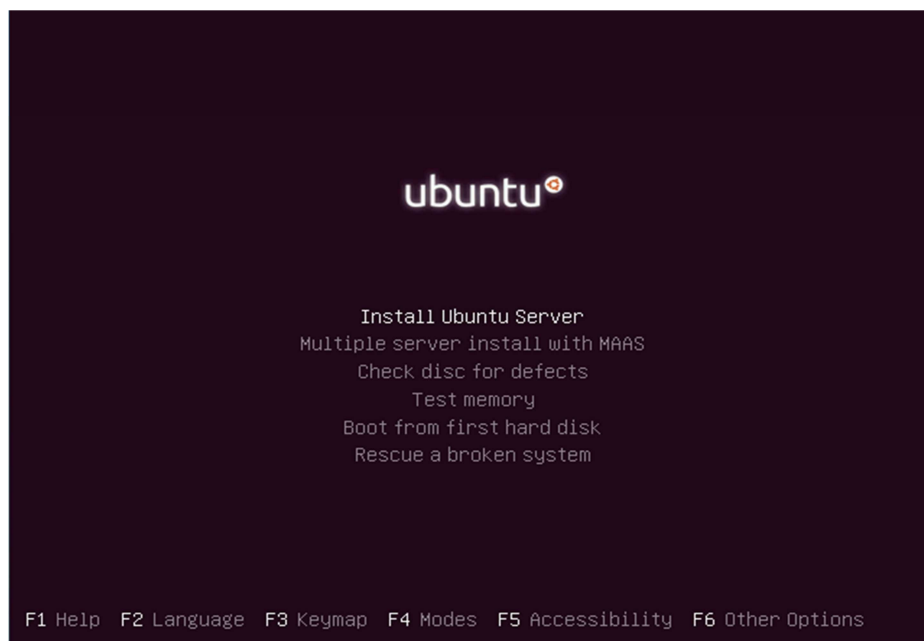
N.B – When copying and pasting commands please check that symbols and character returns are correctly copied across.

Installing Ubuntu (Minimal Headless System)

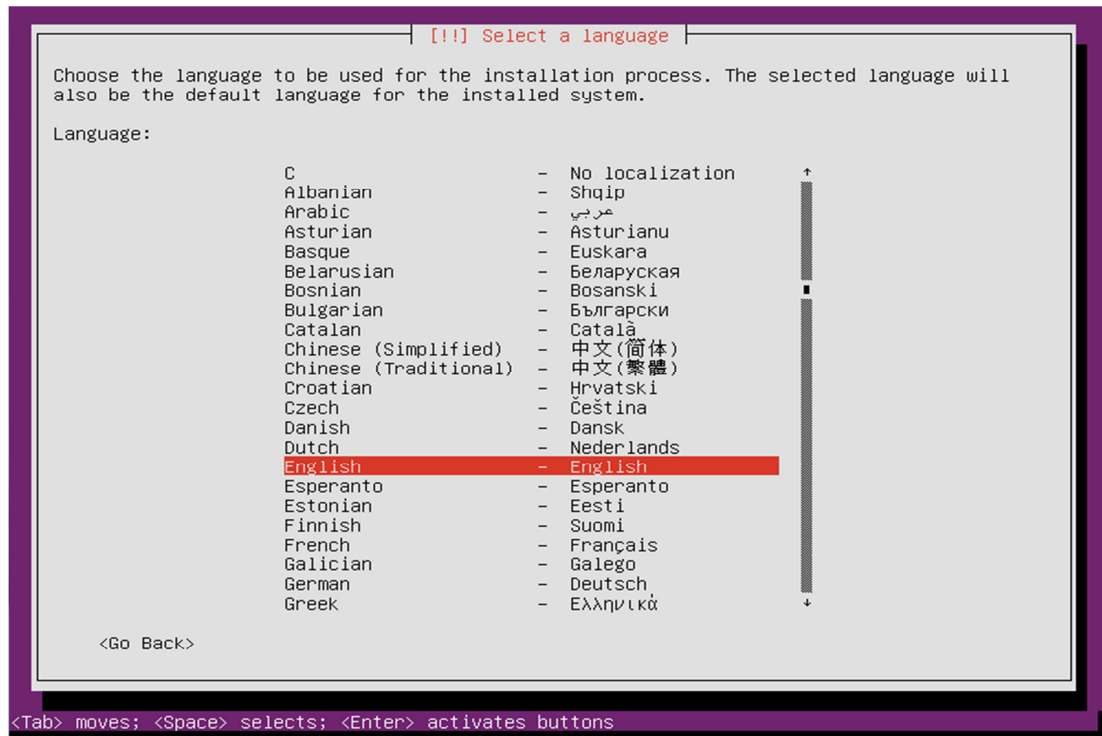
Step 1: - Base System



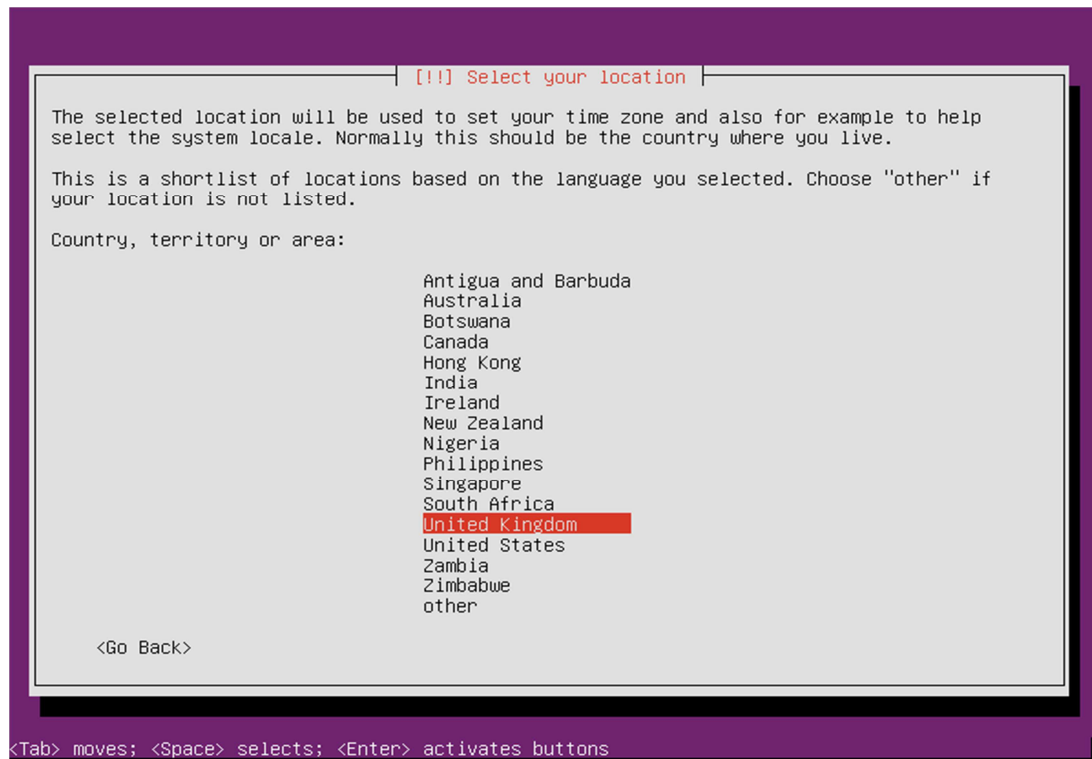
Choose "Install Ubuntu Server"



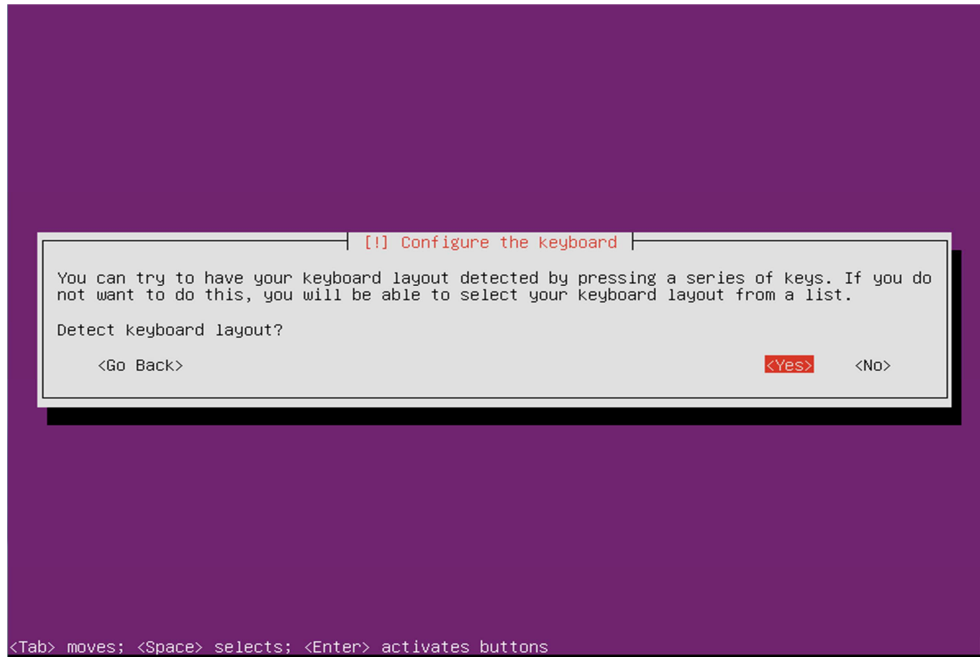
Choose English



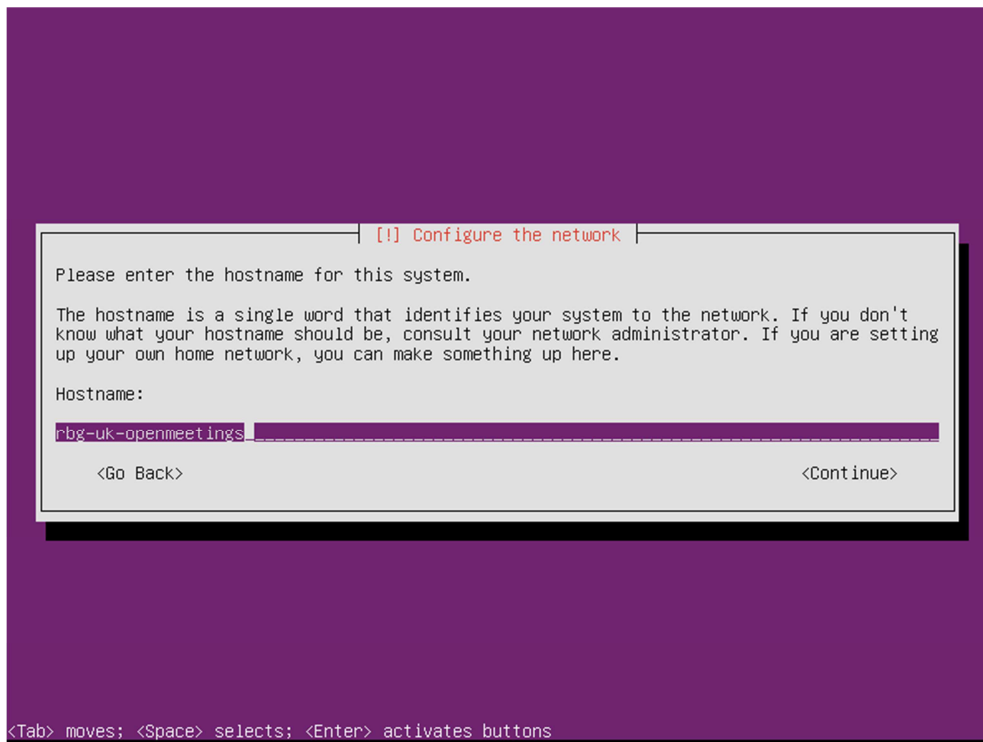
Choose "United Kingdom"



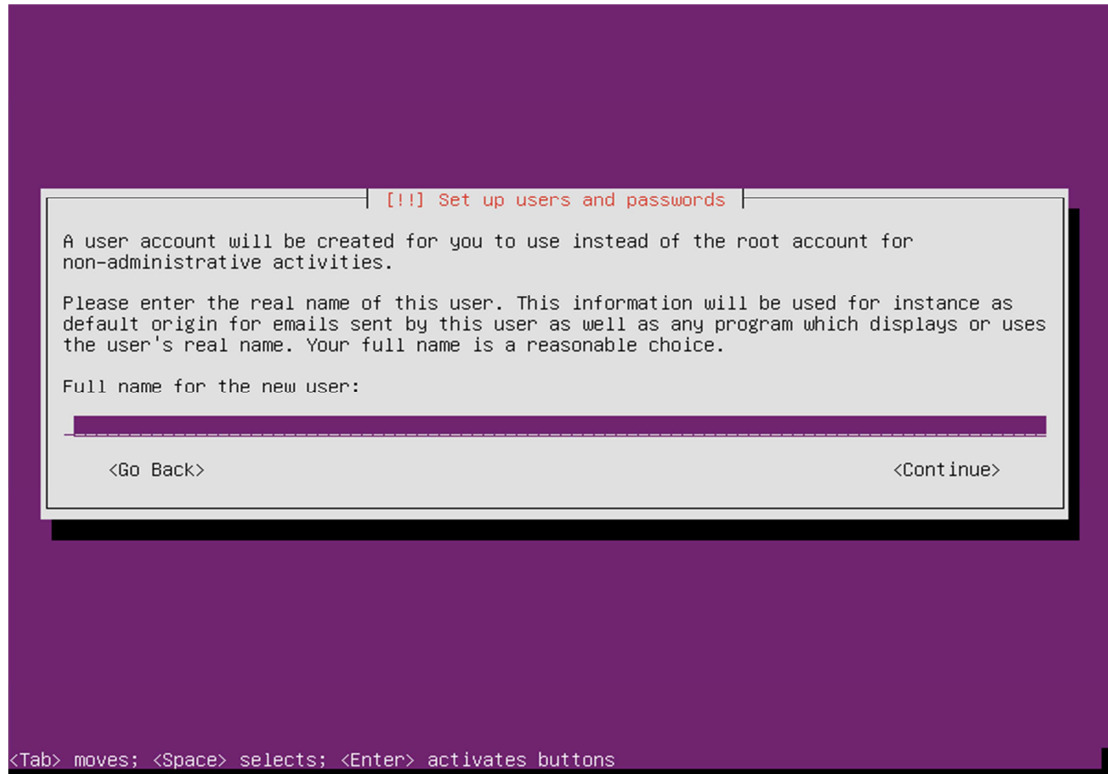
Choose yes to automatically detect keyboard.



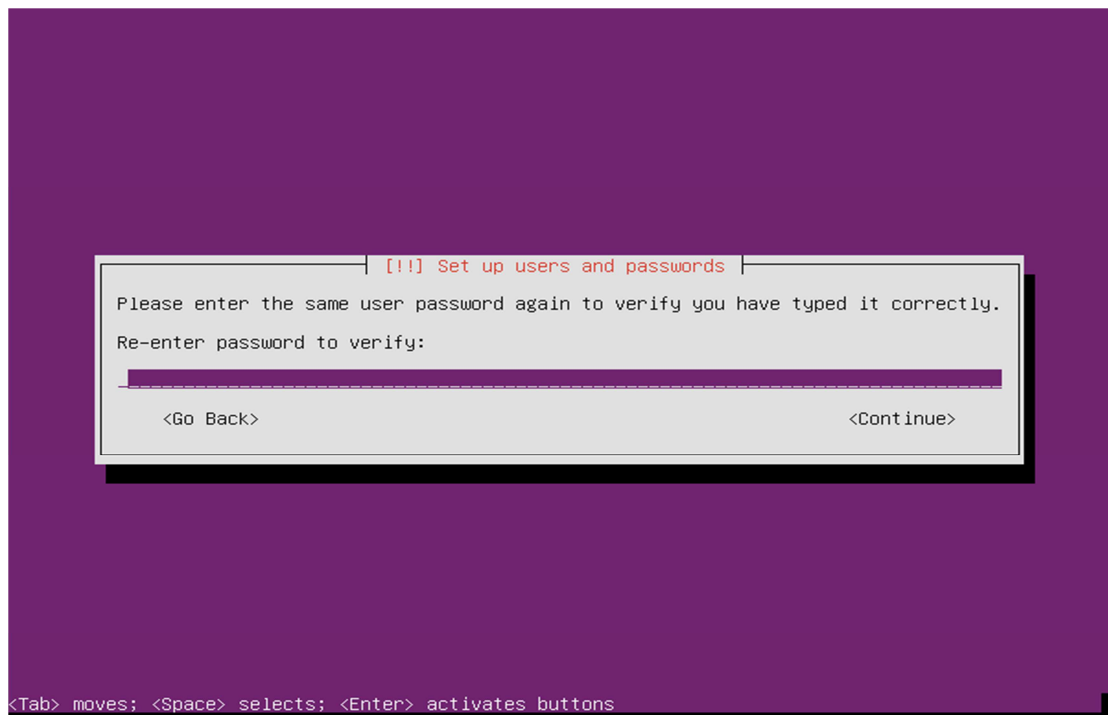
Enter a hostname.



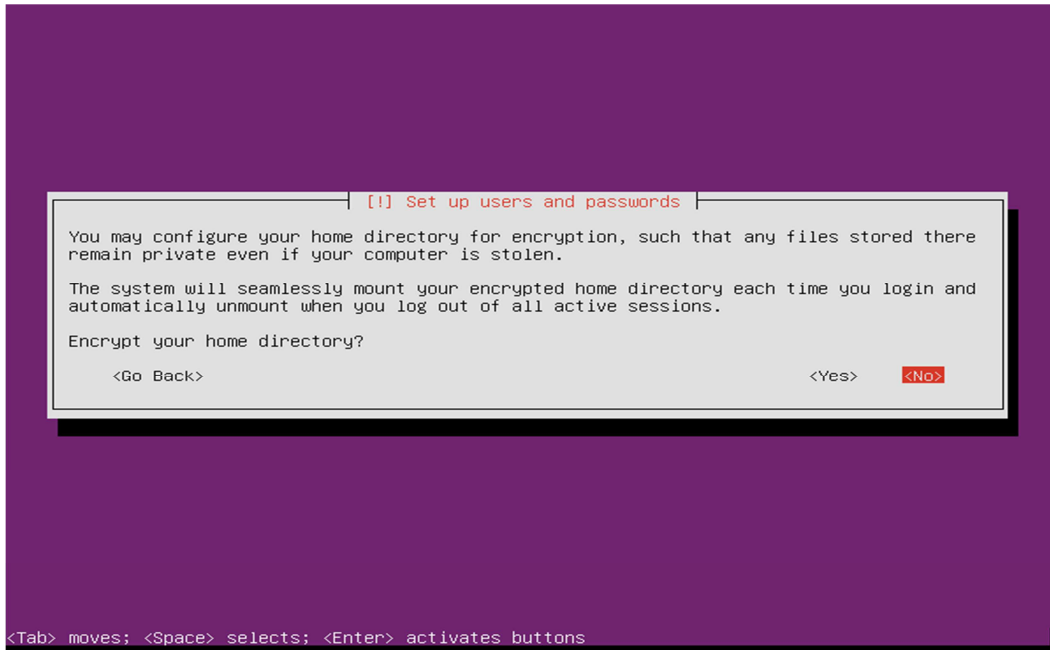
Enter Username.



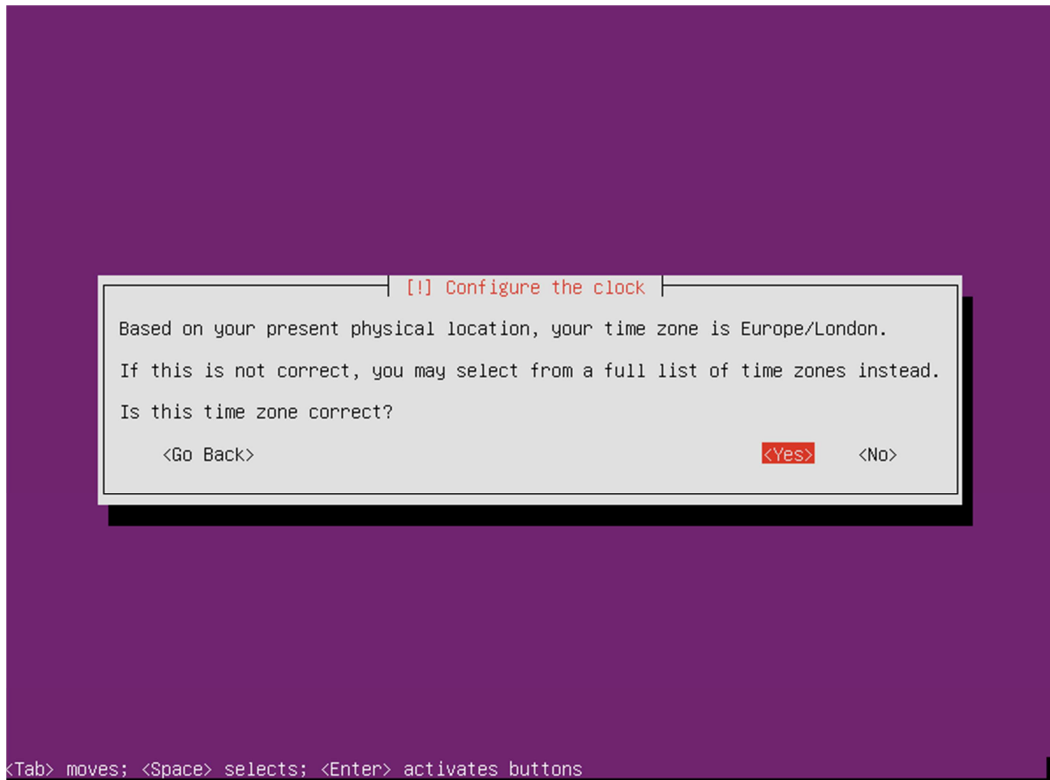
Enter a password for the new user.



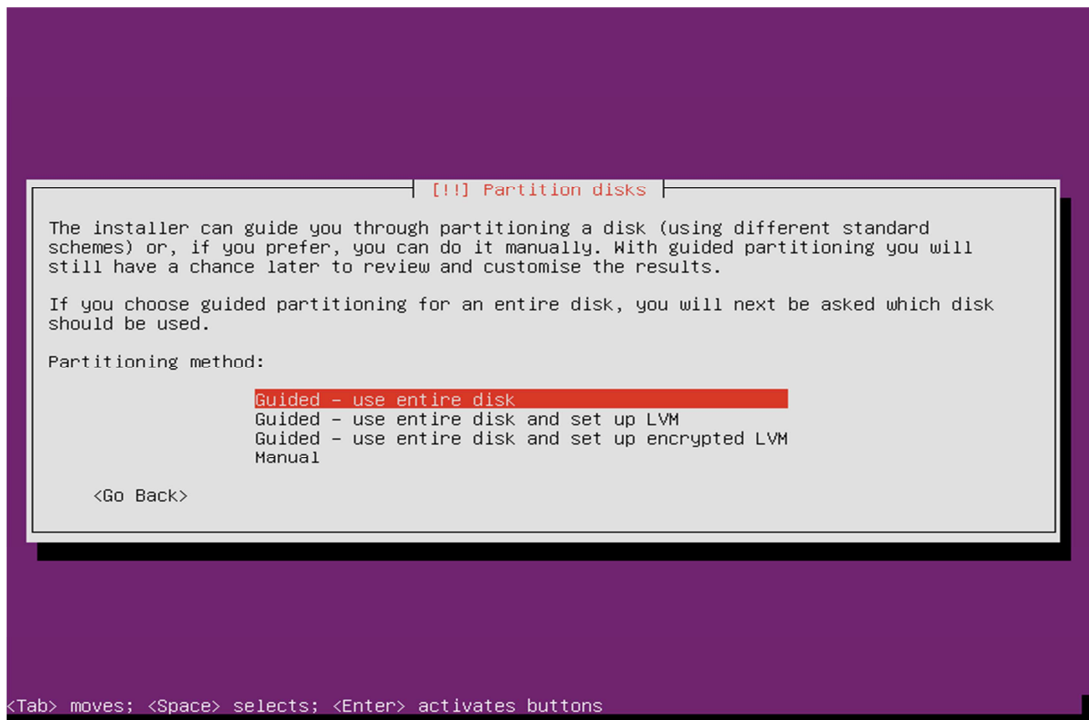
Choose “no” to encrypt the Home Directory.



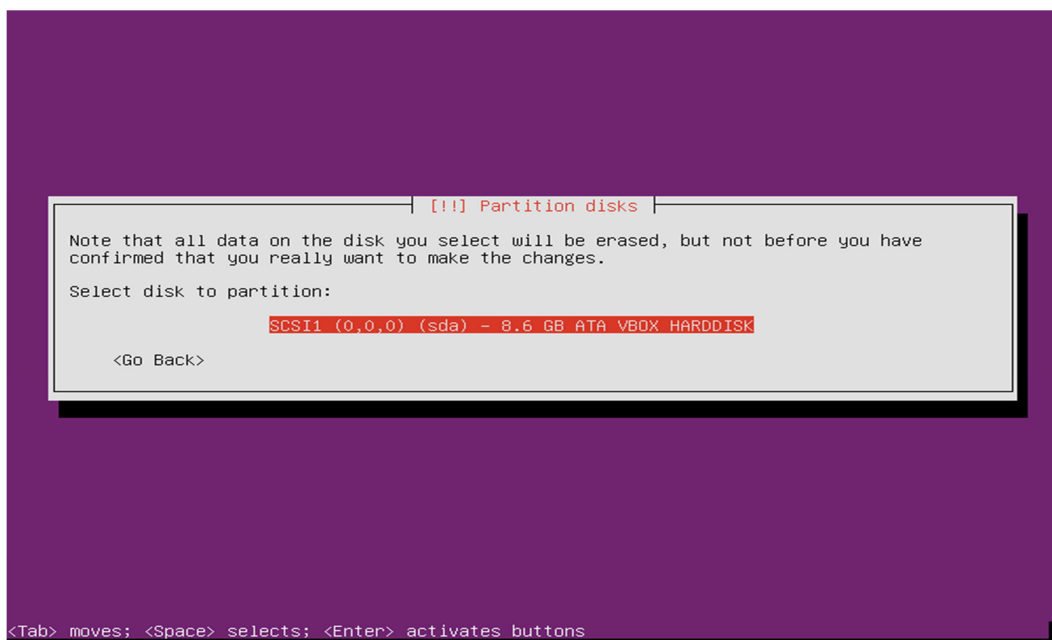
Choose yes to accept the detected time-zone.



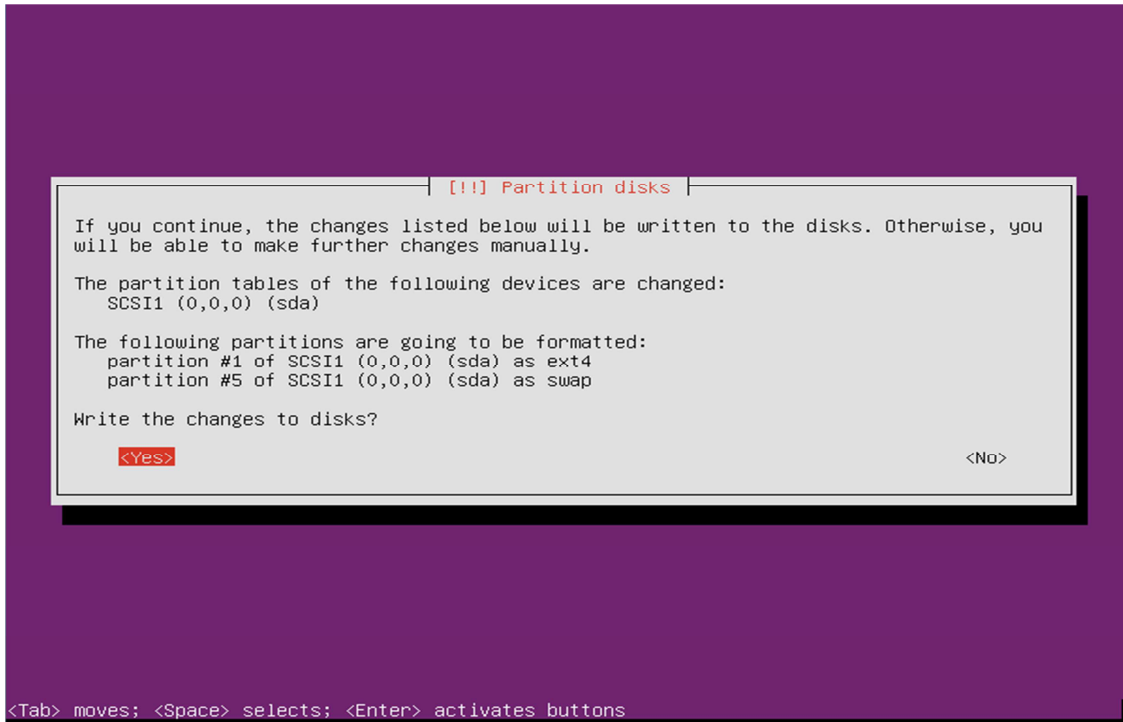
Select "Guided – Use entire disk"



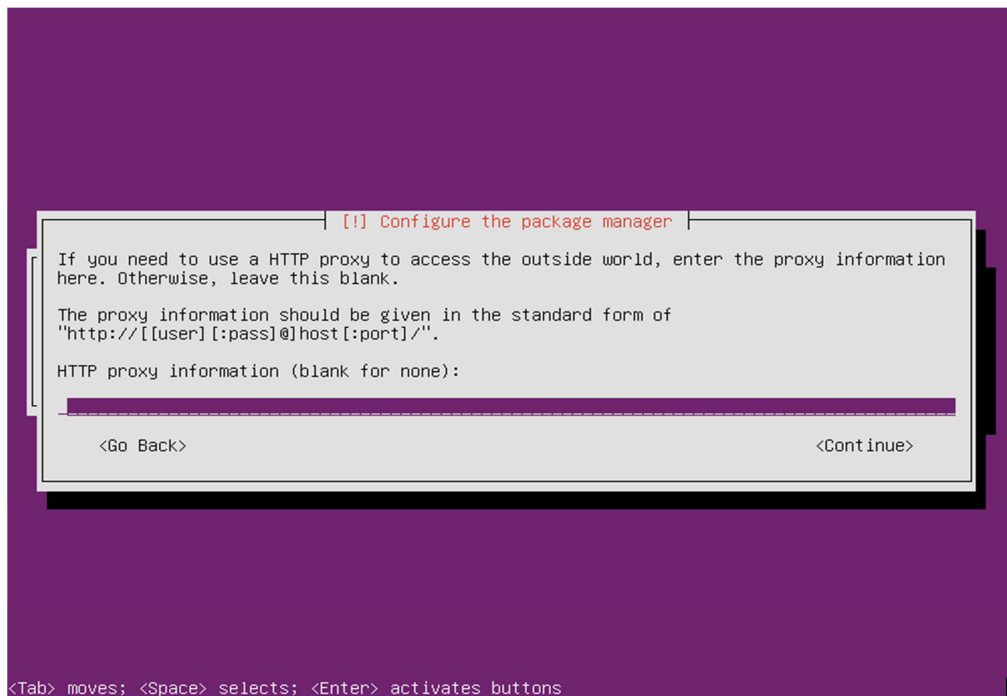
Accept the Disk selection to partition.



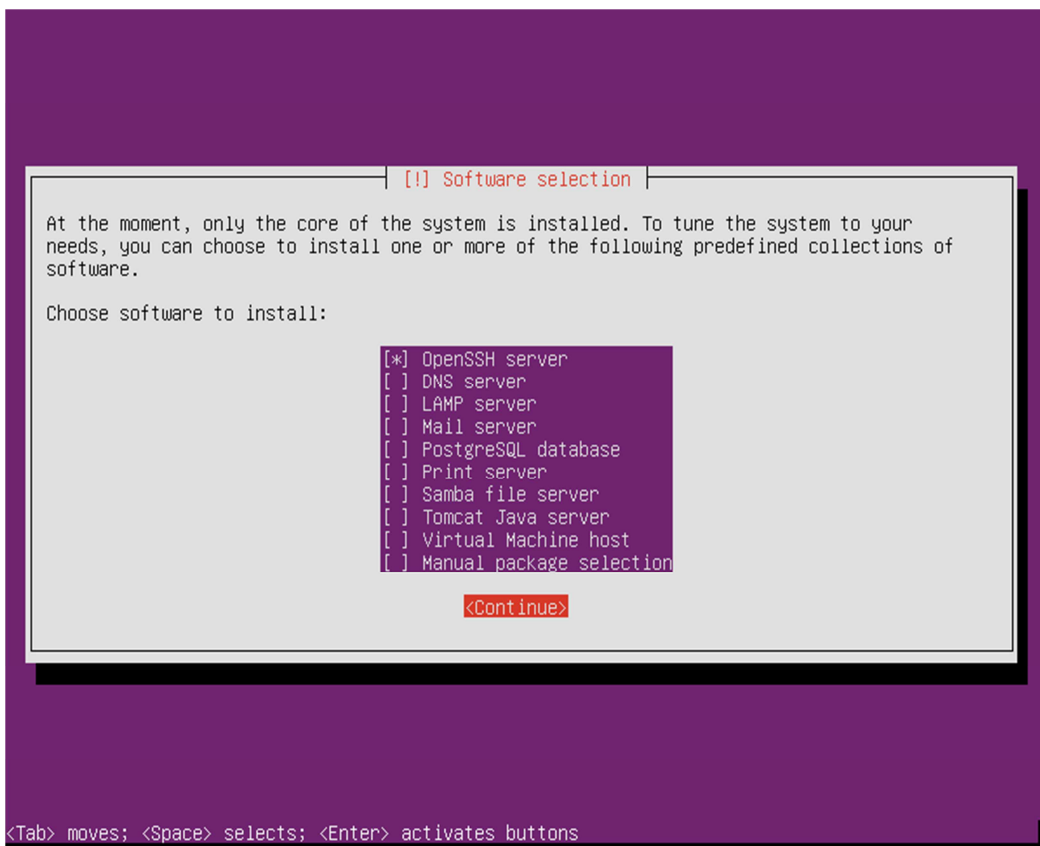
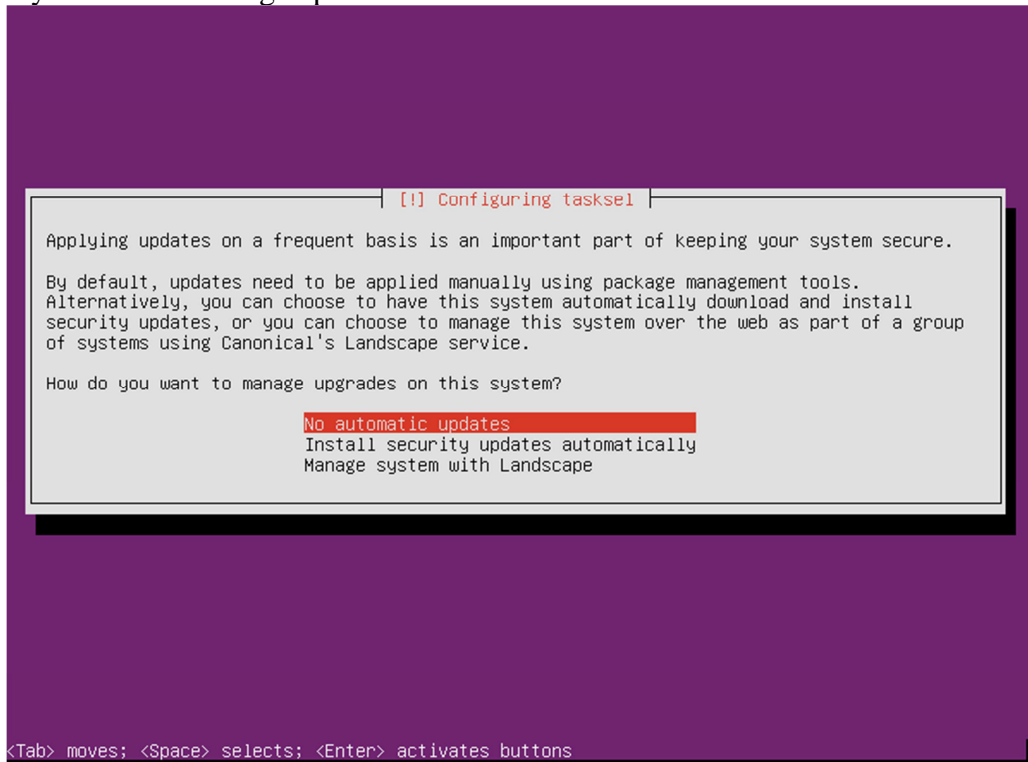
Choose “yes” to accept the changes to disk.



If you use a proxy server please enter that here, in most cases this is not needed and you can simply press enter to continue.

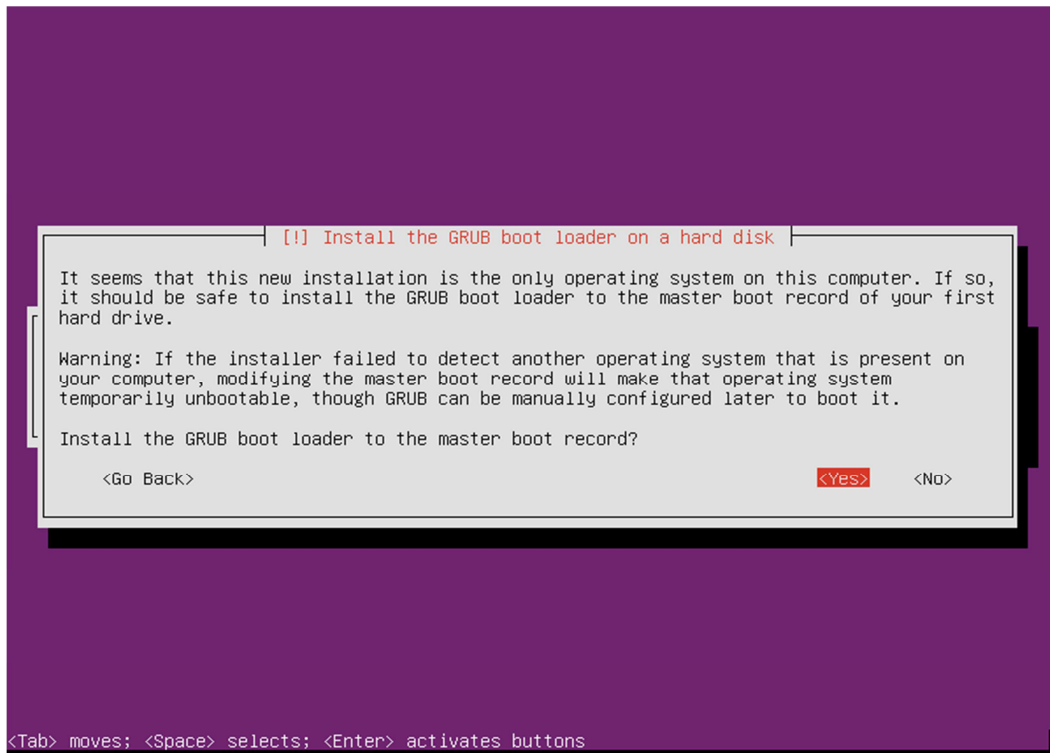


Select how you want to manage updates.

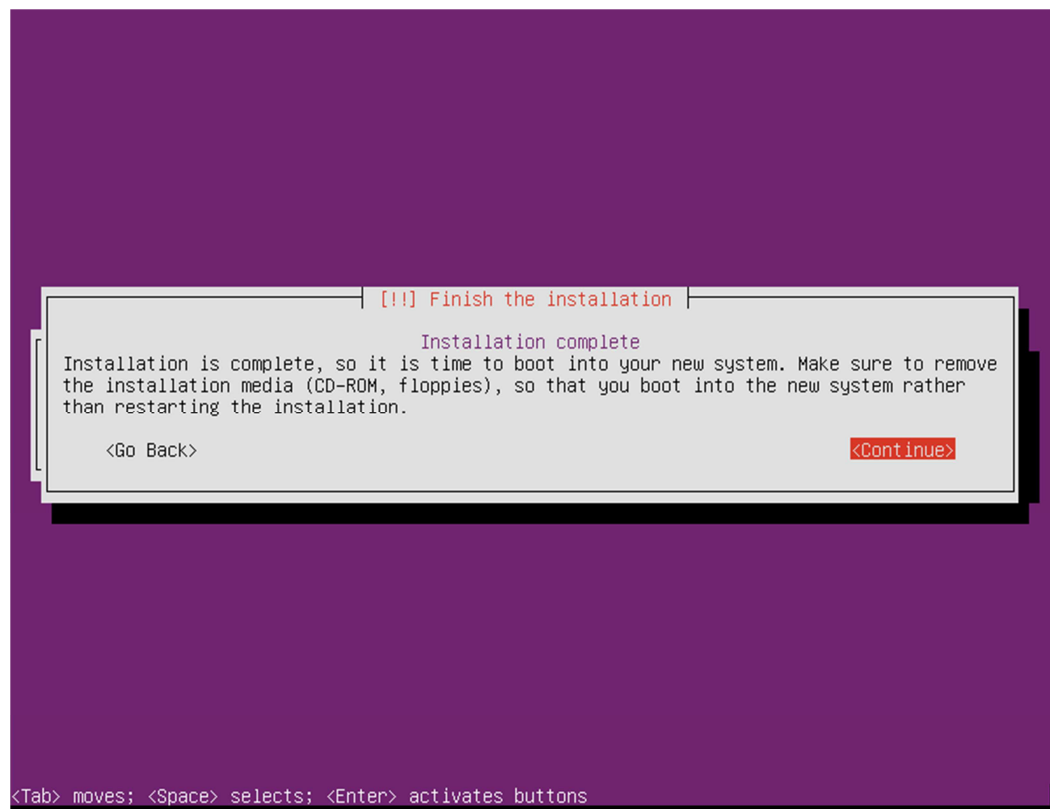


Select OpenSSH and leave the rest blank, then click on Continue.

Choose yes to install and configure the Grub Boot loader.



Press continue to reboot your system.

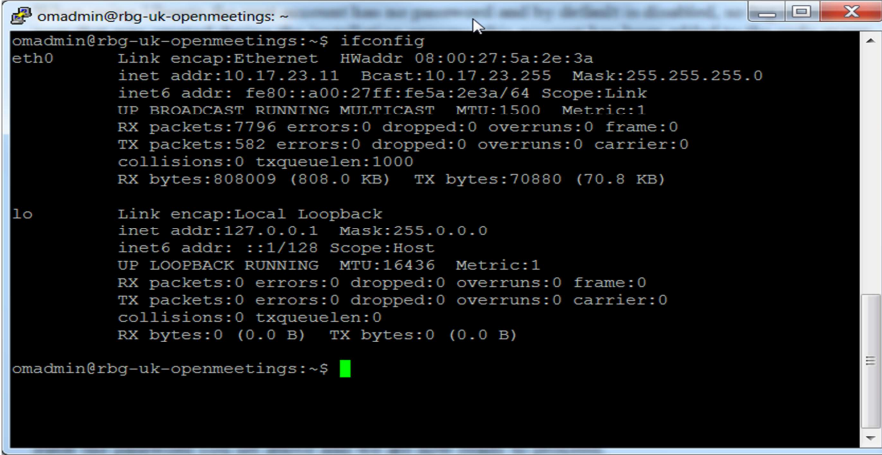


Step 2: - Setup SSH Environment

You should now be at the following screen, the next steps are easier done from a remote desktop using an SSH client such as putty. – But first we need to know our IP address, in most cases this was issued by your DHCP server (unless you specified manual network setup during install) To find your IP address, first logon to your physical machine using root, then issue the following command:

ifconfig

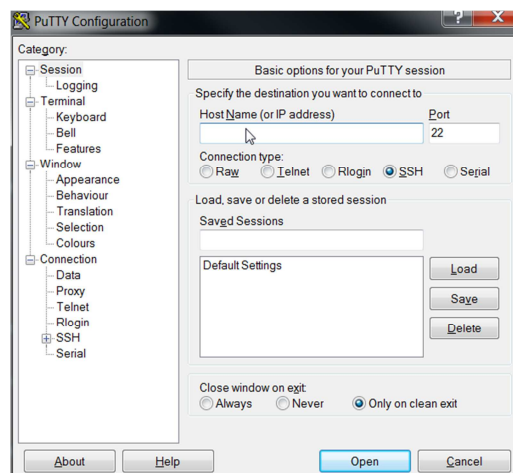
This will show the following screen:



```
omadmin@rbg-uk-openmeetings: ~  
omadmin@rbg-uk-openmeetings:~$ ifconfig  
eth0      Link encap:Ethernet  HWaddr 08:00:27:5a:2e:3a  
          inet addr:10.17.23.11  Bcast:10.17.23.255  Mask:255.255.255.0  
          inet6 addr: fe80::a00:27ff:fe5a:2e3a/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:7796 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:582 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:808009 (808.0 KB)  TX bytes:70880 (70.8 KB)  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
          UP LOOPBACK RUNNING  MTU:16436  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)  
  
omadmin@rbg-uk-openmeetings:~$
```

You can see the IP Address in this case is 10.17.23.11 (Interface eth0)

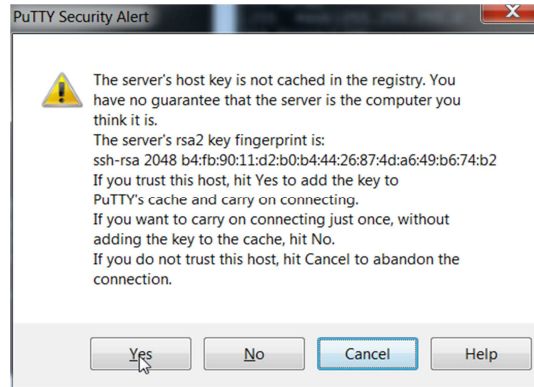
You can now log off of the server.



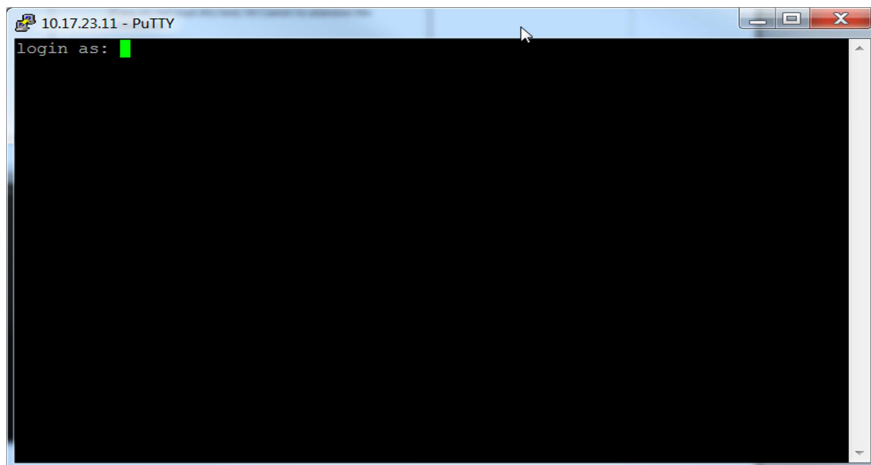
From your desktop machine open your SSH client, in this case we will be using the putty client to connect to our new Server.

Enter the details and choose open.

The first log on you will receive this message; you can choose yes here and accept the key.



And finally this screen:



When using Ubuntu the root account has no password and by default is disabled, so first log on with the user that was created during the installation process (In this case omadmin), this account by default is part of the Admin group and therefore is also part of the sudoers group already, from here we can re-enable the root account.

To do this follow these steps

sudo passwd

Enter the omadmin password first

[sudo] password for omadmin: xxxxxx

Then enter the new root password twice

Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully

We can now enter the following to change user to root

```
su -
```

Enter the password you set above and we are now ready to proceed.

Step 3: Install Dependent software

First update the repos:

```
apt-get update
```

Now let's create our work area

```
mkdir -p /usr/adm
```

We need to first install Libre Office:

```
apt-get install libreoffice -y
```

Libre Office installs OpenJDK so we need to install Oracle Java 6 JDK and update the alternatives – grab the relevant binary from oracle.

```
cd /usr/adm
```

```
wget --no-cookies --header "Cookie: gpw_e24=http%3A%2F%2Fwww.oracle.com%2F" \
http://download.oracle.com/otn-pub/java/jdk/6u32-b05/jdk-6u32-linux-x64.bin
```

Now issue the following to install it and correct the default java.

```
cd /usr/adm
chmod +x jdk-6u32-linux-x64.bin
./jdk-6u32-linux-x64.bin
mkdir -p /usr/lib/jvm
mv jdk1.6.0_32 /usr/lib/jvm/
```

```
update-alternatives --install /usr/bin/javac javac /usr/lib/jvm/jdk1.6.0_32/bin/javac 1
update-alternatives --install /usr/bin/java java /usr/lib/jvm/jdk1.6.0_32/bin/java 1
update-alternatives --install /usr/bin/javaws javaws /usr/lib/jvm/jdk1.6.0_32/bin/javaws 1
```

```
update-alternatives --config javac
update-alternatives --config java
update-alternatives --config javaws
```

Check java by issuing the following

java -version

```
java version "1.6.0_32"
```

```
Java(TM) SE Runtime Environment (build 1.6.0_32-b05)  
Java HotSpot(TM) 64-Bit Server VM (build 20.7-b02, mixed mode)
```

ls -la /etc/alternatives/java*

and confirm the symbolic links point to the correct location.

The last stage of this step is to install the required dependencies for the OM install.

apt-get update

```
apt-get install autoconf automake build-essential checkinstall git libass-dev libfaac-dev -y  
apt-get install libgpac-dev libmp3lame-dev libopencore-amrnb-dev libopencore-amrwb-dev -y  
apt-get install librtmp-dev libspeex-dev -y  
apt-get install libtheora-dev libtool libvorbis-dev pkg-config texi2html zlib1g-dev -y  
apt-get install imagemagick sox libtool -y  
apt-get install libgif-dev xpdf libfreetype6 libfreetype6-dev libjpeg62 libjpeg8 -y  
apt-get install libjpeg8-dev libjpeg-dev libdirectfb-dev -y  
apt-get install libart-2.0-2 libt1-5 zip unzip bzip2 subversion git-core checkinstall -y  
apt-get install texi2html libfaac-dev libfaad-dev libmp3lame-dev libsdl1.2-dev libx11-dev -y  
apt-get install libxfixes-dev libxvidcore-dev zlib1g-dev libogg-dev sox libvorbis0a libvorbis-dev -y  
apt-get install libgsm1 libgsm1-dev libfaad2 flvtool2 lame make g++ -y
```

Step 4: Compile and Install SWFTools 2013-02-19-1826

```
cd /usr/adm  
wget http://www.swftools.org/swftools-2013-02-19-1826.tar.gz  
tar -zxvf swftools-2013-02-19-1826.tar.gz  
cd swftools-2013-02-19-1826/  
./configure  
make  
checkinstall
```

Once that has completed you can now test it by issuing the following:

pdf2swf --version

Which should give you the following output:

```
pdf2swf - part of swftools 2013-02-19-1826
```

Step 5: Install yasm

```
cd /usr/adm
wget http://www.tortall.net/projects/yasm/releases/yasm-1.2.0.tar.gz
tar xzvf yasm-1.2.0.tar.gz
cd yasm-1.2.0
./configure
make
checkinstall --pkgname=yasm --pkgversion="1.2.0" --backup=no --deldoc=yes \
--fstrans=no --default
```

Step 6: Install x264

```
cd /usr/adm
git clone --depth 1 git://git.videolan.org/x264.git
cd x264
./configure --enable-static
make
checkinstall --pkgname=x264 --pkgversion="3:${./version.sh | awk -F'[ ' ]' \
'/POINT/{print $4"+git"$5}')" --backup=no --deldoc=yes --fstrans=no --default
```

Step 7: Install AAC audio encoder

```
cd /usr/adm
git clone --depth 1 git://github.com/mstorsjo/fdk-aac.git
cd fdk-aac
autoreconf -fiv
./configure --disable-shared
make
checkinstall --pkgname=fdk-aac --pkgversion="$(date +%Y%m%d%H%M)-git" \
--backup=no --deldoc=yes --fstrans=no --default
```

Step 8: Install VP8 video encoder and decoder

```
cd /usr/adm
git clone --depth 1 http://git.chromium.org/webm/libvpx.git
cd libvpx
./configure --disable-examples --disable-unit-tests
make
checkinstall --pkgname=libvpx --pkgversion="1:(date +%Y%m%d%H%M)-git" \
--backup=no --deldoc=yes --fstrans=no --default
```

Step 9: Compile and Install ffmpeg 0.11.1

Let's go back to our temporary working area

```
cd /usr/adm
```

Download, compile and install ffmpeg by issuing these commands:

```
wget http://ffmpeg.org/releases/ffmpeg-1.1.3.tar.gz
tar -zxvf ffmpeg-1.1.3.tar.gz
cd ffmpeg-1.1.3
```

```
./configure --enable-gpl --enable-libass --enable-libfaac --enable-libfdk-aac \
--enable-libmp3lame --enable-libopencore-amrnb --enable-libopencore-amrwb \
--enable-libspeex --enable-librtmp --enable-libtheora --enable-libvorbis \
--enable-libvpx --enable-x11grab --enable-libx264 --enable-nonfree --enable-version3 \
--enable-libxvid --enable-libgsm
make
checkinstall --pkgname=ffmpeg --pkgversion="'7:$(date +%Y%m%d%H%M)-git" \
--backup=no --deldoc=yes --fstrans=no --default
```

Once that has completed you can now test it by issuing the following:

```
ffmpeg -version
```

Which should give you the following output:

```
ffmpeg version 0.11.1
```

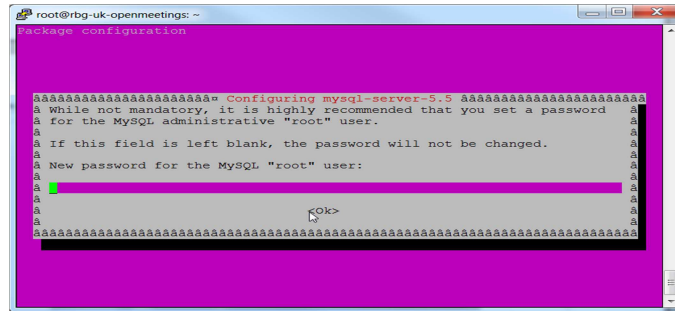
Step 10: Install qt-faststart

```
cd /usr/adm/ffmpeg-1.1.3/
make tools/qt-faststart
checkinstall --pkgname=qt-faststart --pkgversion="'$(date +%Y%m%d%H%M)-git" \
--backup=no --deldoc=yes --fstrans=no --default install -Dm755 tools/qt-faststart \
/usr/local/bin/qt-faststart
```


Step 11: - Create mysql DB for OM

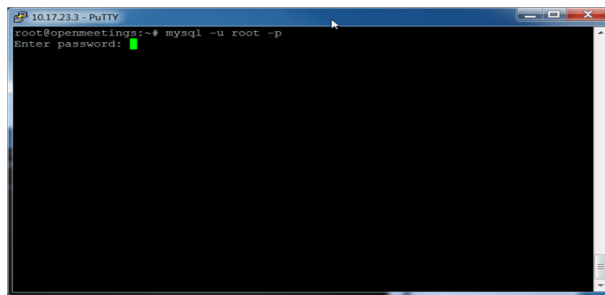
Now we need to install MYSQL, issue this command (In this case username and password are openmeetings : ompassword)

apt-get install mysql-server -y



Enter the password as before “ompassword” and choose ok.
Now let’s create the needed DB’s for OM 2.x, issue the following commands:

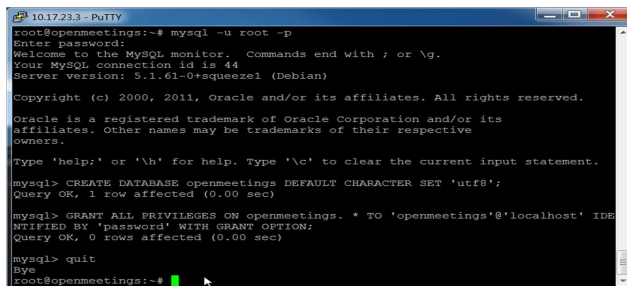
mysql -u root -p



Enter password “ompassword”

Now issue these: (Assuming username openmeetings and password = password)

**CREATE DATABASE openmeetings DEFAULT CHARACTER SET 'utf8';
GRANT ALL PRIVILEGES ON openmeetings.* TO 'openmeetings'@'localhost'
IDENTIFIED BY 'password' WITH GRANT OPTION;
quit**



Successful DB creation shown above.

Step 7: Install JOD Converter

Let's go back to our temporary working area

```
cd /usr/adm
```

Download, extract JOD by issuing these commands: **(We will move the JOD location after the installation of OM 2.x)**

```
wget http://jodconverter.googlecode.com/files/jodconverter-core-3.0-beta-4-dist.zip
unzip jodconverter-core-3.0-beta-4-dist.zip
```

Step 8: Install ANT 1.8.4 for compiling latest OM 2.x

Let's go back to our temporary working area

```
cd /usr/adm
```

Download, extract ANT by issuing these commands:

```
wget http://mirror.catn.com/pub/apache//ant/binaries/apache-ant-1.8.4-bin.tar.gz
tar -zxvf apache-ant-1.8.4-bin.tar.gz
```

Once that has completed you can test it by issuing the following commands:

```
cd /usr/adm/apache-ant-1.8.4/bin
./ant -version
```

This should output the following:

```
Apache Ant(TM) version 1.8.4 compiled on May 22 2012
```

Step 9: Download and compile latest OM 2.1

Again back to our working area:

```
cd /usr/adm
```

Then check out the latest source code using the following:

```
svn checkout https://svn.apache.org/repos/asf/openmeetings/trunk/singlewebapp/
```

Once that has completed we can then build the source by issuing the following:

```
cd /usr/adm/singlewebapp  
/usr/adm/apache-ant-1.8.4/bin/ant clean.all  
/usr/adm/apache-ant-1.8.4/bin/ant -Ddb=mysql
```

This will take a little while depending on your system, once it has finished you should be left the following message:

```
BUILD SUCCESSFUL
```

Step 9a: Install pre-built OM 2.x (Alternative to Step 9)

Download the latest build from the following links:

2.0

<https://builds.apache.org/view/M-R/view/OpenMeetings/job/OpenMeetings%202.0/lastSuccessfulBuild/artifact/2.0/dist/>

Or

2.1

<https://builds.apache.org/job/openmeetings/lastSuccessfulBuild/artifact/singlewebapp/dist/>

The file will be something like the following “apache-openmeetings-incubating-2.xxxxx.tar.gz: (Where xxx is the date and build version)

We can do this using wget, so first we need to go back to our build area like so:

```
cd /usr/adm  
mkdir -p singlewebapp/dist/red5  
cd singlewebapp/dist/red5
```

Then grab the file and extract it:

```
wget https://builds.apache.org/job/openmeetings/lastSuccessfulBuild/artifact/singlewebapp/dist/apache-openmeetings-incubating-2.xxxxx.tar.gz
```

```
tar -zxvf apache-openmeetings-incubating-2.xxxxx.tar.gz
```

Now download the mysql connector from here:

```
http://www.mysql.com/downloads/connector/j/
```

```
cd /usr/adm/singlewebapp/dist/red5/webapps/openmeetings/WEB-INF/lib
```

```
wget http://ftp.up.ac.za/pub/linux/mysql/Downloads/Connector-J/mysql-connector-java-5.1.20.zip
```

```
unzip mysql-connector-java-5.1.20.zip
```

```
cd mysql-connector-java-5.1.20
```

```
mv mysql-connector-java-5.1.20-bin.jar  
/usr/adm/singlewebapp/dist/red5/webapps/openmeetings/WEB-INF/lib
```

Step 10: Install compiled\Pre-Built OM 2.x

Now we need to move the compiled source into the correct location, in this system we are using /usr/lib/red5, so issue the following commands to move the root folder over:

```
cd /usr/adm/singlewebapp/dist  
mv red5/ /usr/lib/
```

Let's move the JOD into place now

```
cp -R /usr/adm/jodconverter-core-3.0-beta-4 /usr/lib/red5/webapps/openmeetings
```

And set some permissions and ownerships

```
chown -R nobody /usr/lib/red5  
chmod +x /usr/lib/red5/red5.sh  
chmod +x /usr/lib/red5/red5-debug.sh
```

Set the start-up script for OM 2.x by issuing the following:

```
vi /etc/init.d/red5
```

and adding the following:

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:      red5
# Required-Start: $remote_fs $syslog
# Required-Stop:  $remote_fs $syslog
# Default-Start:  2 3 4 5
# Default-Stop:   0 1 6
# Short-Description: Starts red5 server for Openmeetings.
### END INIT INFO
# For RedHat and cousins:
# chkconfig: 2345 85 85
# description: Red5 flash streaming server for OpenMeetings
# processname: red5
# Created By: Sohail Riaz (sohaileo@gmail.com)
# Modified by Alvaro Bustos
PROG=red5
RED5_HOME=/usr/lib/red5
DAEMON=$RED5_HOME/$PROG.sh
PIDFILE=/var/run/$PROG.pid
[ -r /etc/sysconfig/red5 ] && . /etc/sysconfig/red5
RETVAL=0
case "$1" in
start)
    cd $RED5_HOME
    start-stop-daemon --start -c nobody --pidfile $PIDFILE \
        --chdir $RED5_HOME --background --make-pidfile \
        --exec $DAEMON >/dev/null 2>/dev/null &
    RETVAL=$?
    if [ $RETVAL -eq 0 ]; then
        echo $! > $PIDFILE
    fi
    echo
;;
stop)
    start-stop-daemon --stop --quiet --pidfile $PIDFILE \
        --name java
    rm -f $PIDFILE
    echo
    [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/$PROG
;;
restart|force-reload)
    $0 stop
    $0 start
;;
status)
    # Debian and Ubuntu 10 status check
    ps aux | grep -f $PIDFILE >/dev/null 2>/dev/null && RETVAL=0 || RETVAL=3
    # Ubuntu 12 status check using improved "start-stop-daemon" status query
    # (use the above command, or comment out above command and uncomment the two below commands.
    # start-stop-daemon --status --pidfile $PIDFILE
    # RETVAL=$?
    [ $RETVAL -eq 0 ] && echo "$PROG is running"
    [ $RETVAL -eq 1 ] && echo "$PROG is not running and the pid file exists"
    [ $RETVAL -eq 3 ] && echo "$PROG is not running"
    [ $RETVAL -eq 4 ] && echo "$PROG - unable to determine status"
;;
checkports)
    netstat -anp | grep soffice
    netstat -anp | grep java
;;
*)
    echo $"Usage: $0 {start|stop|restart|force-reload|status|checkports}"
    RETVAL=1
esac
exit $RETVAL
```

Save the file and then set the permissions like below:

```
chmod +x /etc/init.d/red5
update-rc.d red5 defaults
```

Now we need to move the persistence files so we can connect to mysql, so issue the following:

Make backup copy

```
mv /usr/lib/red5/webapps/openmeetings/WEB-INF/classes/META-INF/
persistence.xml /usr/lib/red5/webapps/openmeetings/WEB-INF/
classes/META-INF/persistence.xml-ori
```

Rename mysql template to persistence.xml

```
mv /usr/lib/red5/webapps/openmeetings/WEB-INF/classes/META-INF/
mysql_persistence.xml /usr/lib/red5/webapps/openmeetings/WEB-INF/
classes/META-INF/persistence.xml
```

Edit the persistence file and add out mysql details, in this case we used “**openmeetings**” and “**password**” – so issue the following:

```
vi /usr/lib/red5/webapps/openmeetings/WEB-INF/classes/META-INF/persistence.xml
```

Then change the following

```
, Username=openmeetings
, Password=password"/>
```

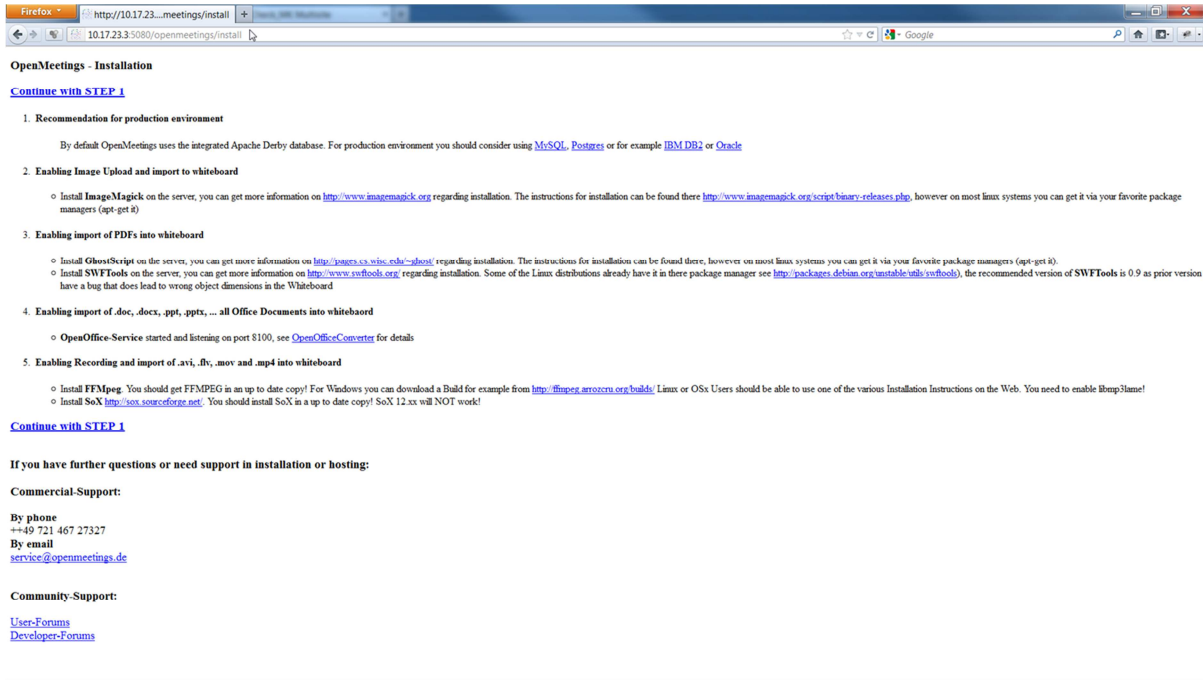
At this stage we are ready to start up OM 2.x for the first time.

```
/etc/init.d/red5 start
```

Now open the browser and go to the following link. **N.B remember to change the IP address to your OM2.x server, the one below 10.17.23.11 is just for this example.**

<http://10.17.23.11:5080/openmeetings/install>

If all went well you should now see this page:



Choose the “Continue with STEP 1” link

OpenMeetings - Installation

Userdata	
Username	<input type="text"/>
Userpass	<input type="text"/>
Email	<input type="text"/>
User Time Zone	New Zealand (Etc/GMT+12 (New Zealand))
Organisation(Domains)	
Name	<input type="text"/>
Configuration	
Allow self-registering (allow_frontend_register)	Yes
Send Email to new registered Users (sendEmailAtRegister)	Yes
New Users need to verify their EMail (sendEmailWithVerificationCode)	Yes
Default Rooms of all types will be created	Yes
Mail-Referer (system_email_addr)	noreply@localhost
SMTP-Server (smtp_server)	localhost
SMTP-Server Port(default Smtip-Server Port is 25) (smtp_port)	25
SMTP-Username (email_userpass)	<input type="text"/>
SMTP-Userpass (email_userpass)	<input type="text"/>
Enable TLS in Mail Server Auth	No
Set inviter's email address as ReplyTo in email invitations (inviter.email.as.replyto)	Yes
Default Language	english

The only section we need to fill out at this stage is the following:

Username: **omadmin**
Userpass: **ompassword**
Email: **something@something.com**
TimeZone: **United Kingdom**
Domain Name: **somedomain**

Now click on INSTALL at the bottom of the page, this will then create all the needed tables etc.. - it can take a little while but be patient.

OpenMeetings - Installation Complete!

[Enter the Application](#)

If your Red5-Server runs on a different Port or on a different domain
[alter the config values of the client](#)

Mailing list

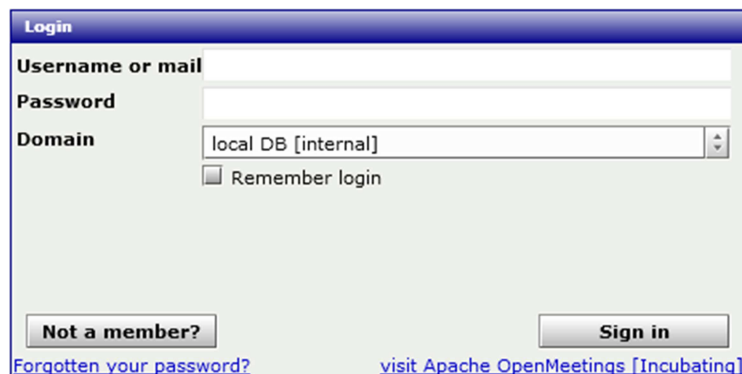
<http://incubator.apache.org/openmeetings/mail-lists.html>

There are some companies that also offer commercial support for Apache OpenMeetings:

<http://incubator.apache.org/openmeetings/commercial-support.html>

Once that has completed you can now enter the application by clicking on the “**Enter the Application**” link

You should see the following logon screen:



Enter these details to sign in.

Username: **omadmin**
Userpass: **ompassword**

Step 11: Add relevant paths to the configuration

Once logged in go to **Administration > Configuration**

You will see on the left hand pane a list of keys and values, the ones we are interested in are

SWFTools Path	/usr/local/bin/
ImageMagick Path	/usr/bin/
FFMPEG Path	/usr/local/bin
SoX Path	/usr/bin/
JOD Path	/usr/lib/red5/webapps/openmeetings/jodconverter-core-3.0-beta-4/lib

Click on the left hand pane option and then enter the value as above, click on the save button to apply the changes; once you have done each key you should see the following:

Apache OpenMeetings [Incubating]

Home | Recordings | Rooms | Administration

0 - 50 of 67

ID	Key	Value
1	crypt_ClassName	org.openmeetings.util.crypt.MD5Imple
2	screen_viewer	4
3	allow_frontend_register	1
4	default_group_id	1
5	default_domain_id	1
6	smtp_server	localhost
7	smtp_port	25
8	system_email_addr	noreply@localhost
9	email_username	
10	email_userpass	
11	mail.smtp.starttls.enabl	0
12	application.name	OpenMeetings
13	default_lang_id	1
14	swftools_zoom	72
15	swftools_jpegquality	85
16	swftools_path	/usr/local/bin
17	imagemagick_path	/usr/bin
18	sox_path	/usr/bin
19	ffmpeg_path	
20	office.path	
21	jod.path	/usr/lib/red5/webapps/openmeetings/j
22	rss_feed1	null
23	rss_feed2	null
24	sendEmailAtRegister	1
25	sendEmailWithVerificatic	1
26	default_export_font	TimesNewRoman
27	default.rpc.userid	1
28	red5sip.enable	no
29	red5sip.room_prefix	400
30	red5sip.exten_context	rooms
31	sip.enable	no
32	sip.realm	
33	sip.port	
34	sip.proxynome	
35	sip.tunnel	
36	sip.codebase	
37	sip.forcetunnel	true
38	sip.openxg.enable	no
39	openxg.wrapper.url	
40	openxg.client.id	
41	openxg.client.secret	
42	openxg.client.domain	

Configuration

Key

Value

Last update

Updated by

Comment

JOD will find open office in this case so we do not need to set the path.

Step 12: Securing OpenMeetings using encryption (Optional)

12.1 - Generating CSR:

We can do this in a few ways, the first way I will show here is simply by generating a CSR and inserting these into OpenMeetings.

Create a new keystore and key, use the same password for both: (Taken from OM Website <http://incubator.apache.org/openmeetings/RTMPSAndHTTPS.html>)

```
keytool -keysize 2048 -genkey -alias red5 -keyalg RSA -keystore red5/conf/keystore
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: <your hostname, e.g demo.openmeetings.de>
What is the name of your organizational unit?
[Unknown]: Dev
What is the name of your organization?
[Unknown]: OpenMeetings
What is the name of your City or Locality?
[Unknown]: Henderson
What is the name of your State or Province?
[Unknown]: Nevada
What is the two-letter country code for this unit?
[Unknown]: US
Is CN=demo.openmeetings.de, OU=Dev, O=OpenMeetings, L=Henderson, ST=Nevada, C=US
correct?
[no]: yes
Enter key password for <red5>
```

Generate a CSR:

```
keytool -certreq -keyalg RSA -alias red5 -file red5.csr -keystore red5/conf/keystore
```

Submit CSR to your CA of choice and receive a signed certificate

Import your chosen CA's root certificate into the keystore (may need to download it from their site - make sure to get the root CA and not the intermediate one)

```
keytool -import -alias root -keystore red5/conf/keystore -trustcacerts -file root.crt
```

(note: you may receive a warning that the certificate already exists in the system wide keystore - import anyway)

Import the intermediate certificate(s) you normally receive with the certificate:

```
keytool -import -alias intermed -keystore red5/conf/ keystore -trustcacerts -file intermediate.crt
```

Import the certificate you received:

```
keytool -import -alias red5 -keystore red5/conf/keystore -trustcacerts -file
demo.openmeetings.de.crt
```

12.2 – Using Existing certs such as wild card certificates instead of generating a new CSR.

First let's go back to our work area:

```
cd /usr/adm/  
mkdir certs  
cd certs/
```

Using WinSCP or equivalent copy your wild card key and cert files: yourdomain.key.pem and yourdomain.cert.pem - **(These should be in PEM format)**

Now issue the following to convert the files to DER format

```
openssl pkcs8 -topk8 -nocrypt -in apache.key.pem -inform PEM -out key.der -outform DER  
openssl x509 -in apache.cert.pem -inform PEM -out cert.der -outform DER
```

Now we need a couple of files to help us import the DER files into the keystore, so issue the following:

```
wget http://www.agentbob.info/agentbob/80/version/default/part/AttachmentData/data/ImportKey.java  
wget http://www.agentbob.info/agentbob/81/version/default/part/AttachmentData/data/ImportKey.class
```

Then use these commands to import:

```
java ImportKey key.der cert.der
```

Finally move the keystore to the correct location

```
mv /root/keystore.ImportKey /usr/lib/red5/conf/keystore
```

N.B = Alias:importkey Password:importkey (When using the java import key files, you can change the password afterwards)

Now that we have either a new Cert of the wild card cert inside our Keystore we need to make some changes to OM 2.x to use these certificates and thus encrypt communications using HTTPS and RTMPS.

To use RTMPS do the following:

First make some changes to the red5-core.xml file by issuing the following:

```
cd /usr/lib/red5/conf
vi red5-core.xml
```

now uncomment `<!-- RTMPS -->` section by removing the `<!--` and the `-->` leaving this:

```
<bean id="rtmpsMinaIoHandler"
  class="org.red5.server.net.rtmps.RTMPSMinaIoHandler">
  <property name="handler" ref="rtmpHandler" />
  <property name="codecFactory" ref="rtmpCodecFactory" />
  <property name="rtmpConnManager" ref="rtmpMinaConnManager" />
  <property name="keyStorePassword" value="{rtmps.keystorepass}" />
  <property name="keystoreFile" value="conf/keystore" />
</bean>

<bean id="rtmpsTransport" class="org.red5.server.net.rtmp.RTMPMinaTransport" init-
method="start" destroy-method="stop">
  <property name="ioHandler" ref="rtmpsMinaIoHandler" />
  <property name="connectors">
    <list>
      <bean class="java.net.InetSocketAddress">
        <constructor-arg index="0" type="java.lang.String" value="{rtmps.host}" />
        <constructor-arg index="1" type="int" value="{rtmps.port}" />
      </bean>
    </list>
  </property>
  <property name="ioThreads" value="{rtmp.io_threads}" />
  <property name="jmxPollInterval" value="1000" />
  <property name="tcpNoDelay" value="{rtmp.tcp_nodelay}" />
</bean>
```

Save this file and then do the following:

```
cd /usr/lib/red5/conf
vi red5.properties
```

```
set rtmps.port=5443
```

```
rtmps.keystorepass=password (password = password you set on your new keystore)
```

Now edit config.xml by doing the following:

```
cd /usr/lib/red5/webapps/openmeetings/
vi config.xml
```

Set these following values:

```
<rtmpsslport>5443</rtmpsslport>  
<useSSL>yes</useSSL>  
<proxyType>best</proxyType>
```

To use HTTPS do the following:

First make a backup of the original jee-container file by doing the following:

```
cd /usr/lib/red5/conf  
mv jee-container.xml jee-container.xml.orig
```

Then rename the SSL jee template

```
mv jee-container-ssl.xml jee-container.xml
```

Now edit the config.xml

```
cd /usr/lib/red5/webapps/openmeetings/  
vi config.xml
```

set

```
<protocol>https</protocol>  
<red5httpport>443</red5httpport>
```

Lastly edit red5.properties by doing the following:

```
cd /usr/lib/red5/conf  
vi red5.properties
```

set

```
https.port=443  
http.port=443
```

Now restart OM using the following:

```
/etc/init.d/red5 restart
```

We can now connect using the following link:

<https://yourdomain/openmeetings>

Step 13: Installing Reverse Proxy using Apache Web Server (Optional)

Another way to secure the OpenMeetings service is to use Apache as a reverse proxy, to do this we need to do the following:

First install Apache2 and enabling relevant modules by running the following commands:

```
apt-get install apache2
a2enmod proxy
a2enmod proxy_http
a2enmod ssl
a2enmod headers
a2enmod rewrite
a2enmod cache
/etc/init.d/apache2 restart
```

We can now redirect port 80 (less secure) or port 443 (secure) to port 5080, to do this we need to create a virtual host, to do this do the following:

```
cd /etc/apache2/sites-enabled/
```

Now for SSL redirect (**using a Cert on Apache instead of keystore**) do the following

```
vi om.yourdomain.com-ssl
```

and add the following

```
<IfModule mod_ssl.c>
#NameVirtualHost *:443
ProxyRequests Off
<VirtualHost *:80>
  ServerAdmin hostmaster@domain.com
  ServerName om.yourdomain.com

ProxyPreserveHost On
RewriteEngine on
```