



arts & culture

Department:
Arts and Culture
REPUBLIC OF SOUTH AFRICA

South African Directory Enquiries

SADE

Installation, build and setup guide

v. 0.1

by

J.W.F. Thirion
Willem Basson

TABLE OF CONTENTS

[TABLE OF CONTENTS](#)
[INTRODUCTION](#)
[REQUIREMENTS](#)
[UBUNTU](#)
[UPDATE AND INSTALL PACKAGES](#)
[CONFIGURE GIT](#)
[CONFIGURE GYP](#)
[INSTALL DEPOT_TOOLS](#)
[INSTALL JAVA](#)
[COMPILE WEBRTC](#)
[INSTALL KALDI](#)
[INSTALL DAHDI](#)
[INSTALL LIBPRI](#)
[INSTALL ASTERISK](#)
[INSTALL UNIMRCP-DEPS](#)
[INSTALL SPHINXBASE](#)
[INSTALL POCKETSPHINX](#)
[INSTALL FLITE](#)
[INSTALL SADE](#)
[INSTALL UNIMRCP](#)
[INSTALL UNIMRCP-SOLUTIONS](#)
[INSTALL ASTERISK-UNIMRCP](#)
[GENERAL](#)
[ADD WEBRTC TO UNIMRCP](#)
[INSTALLATION-SPECIFIC TASKS](#)

1. INTRODUCTION

This document is a step-by-step guide on how to install, build and setup the South African Directory Enquiries system's back-end speech recognition and MRCP server. It assumes only basic Linux skills.

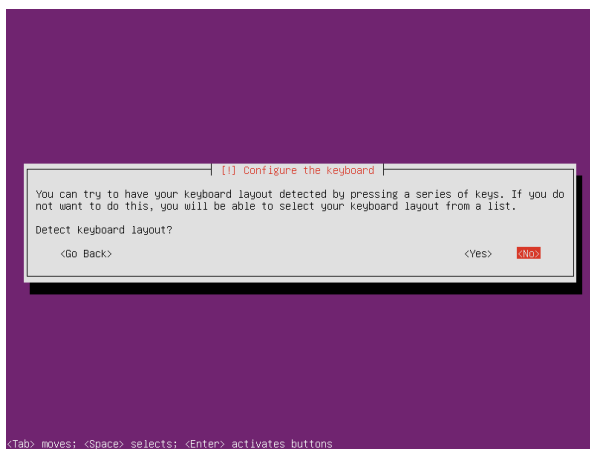
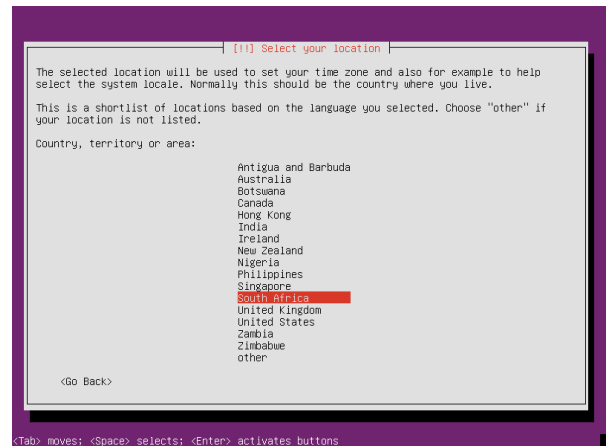
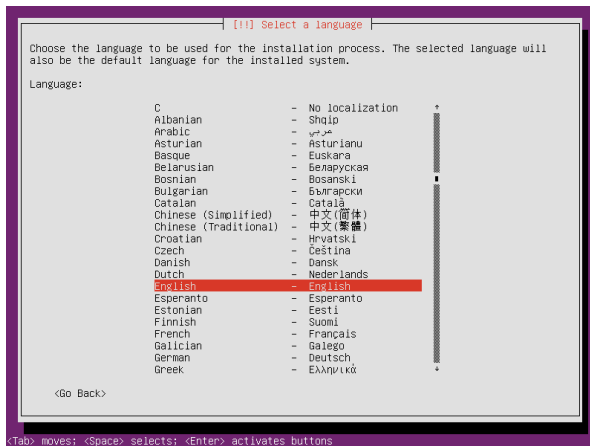
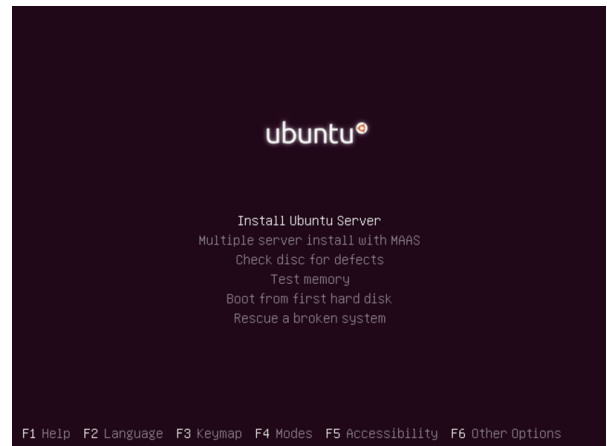
2. REQUIREMENTS

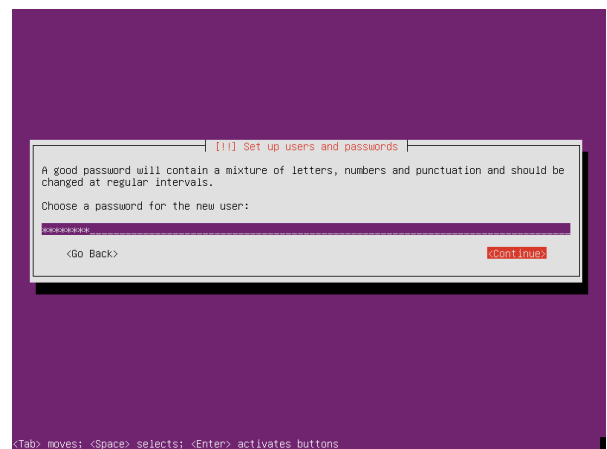
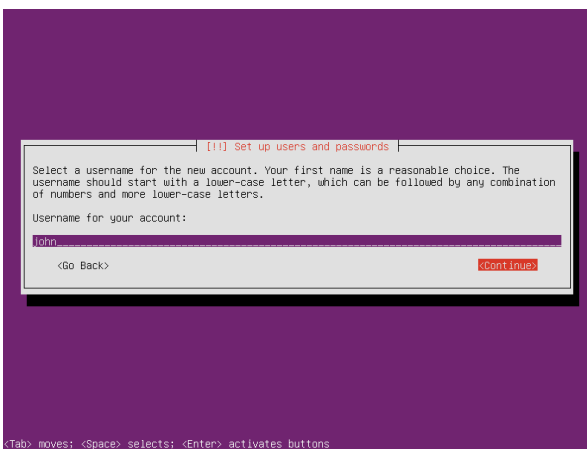
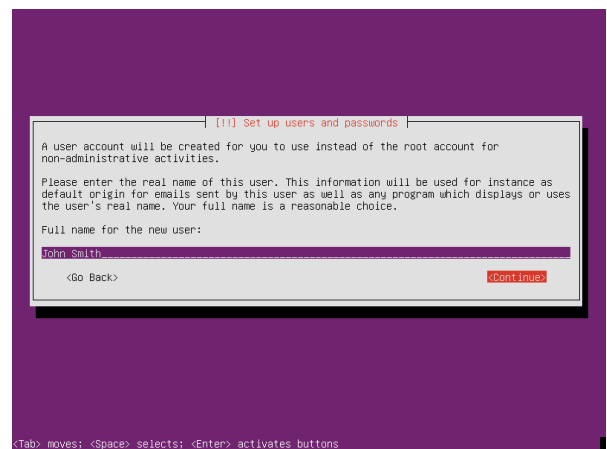
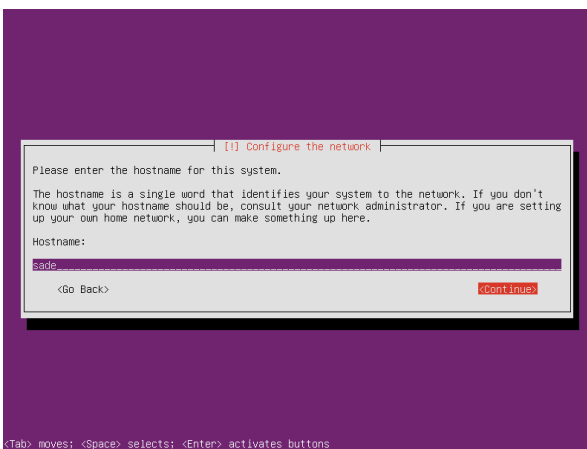
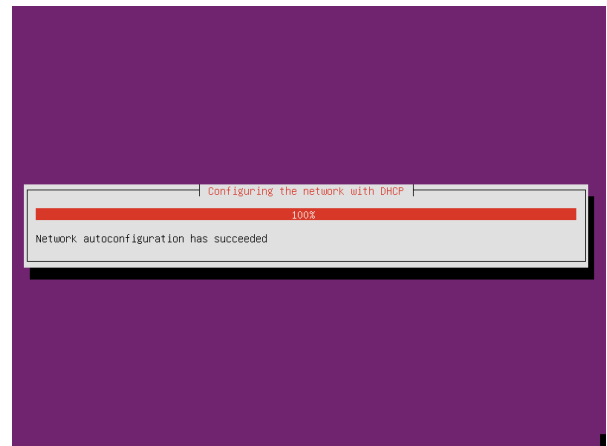
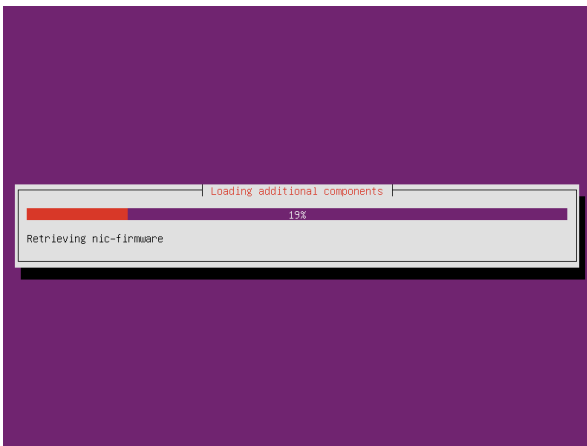
18 GB of free space

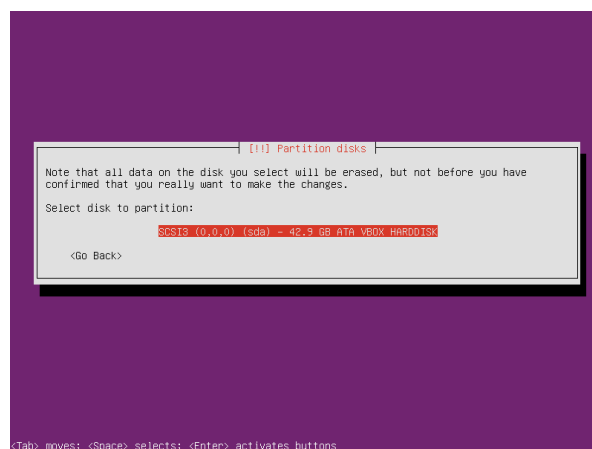
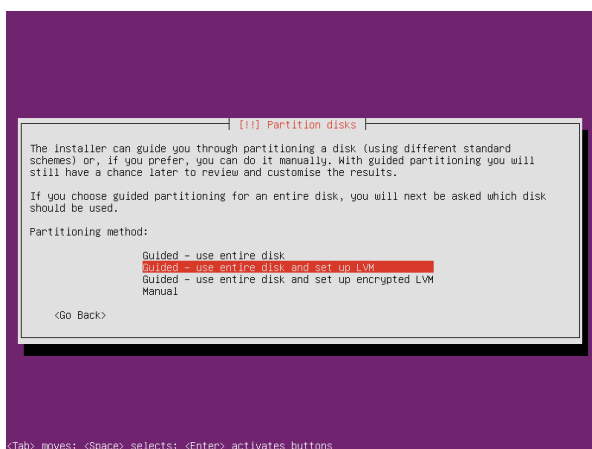
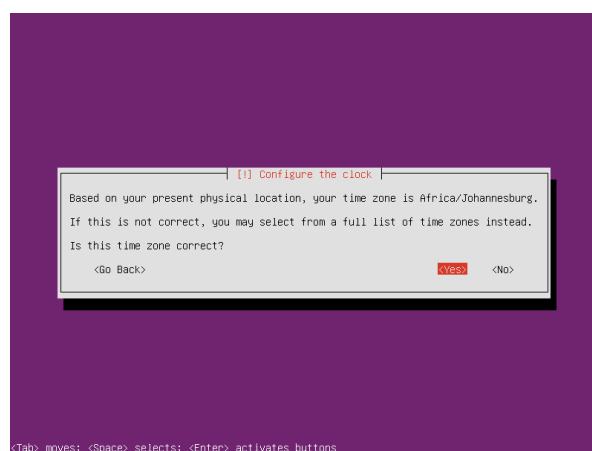
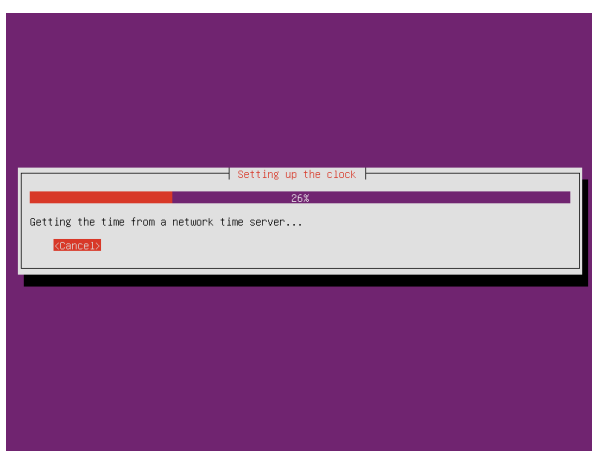
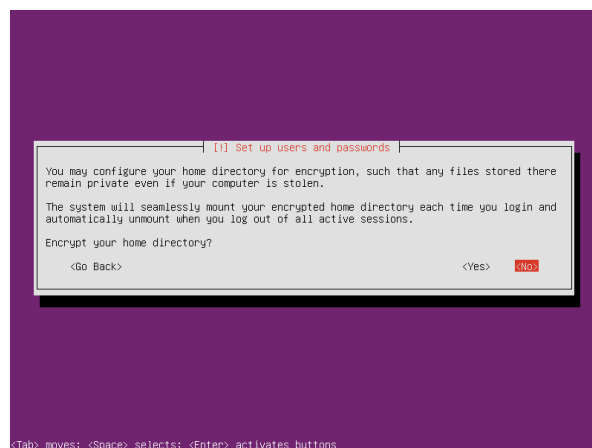
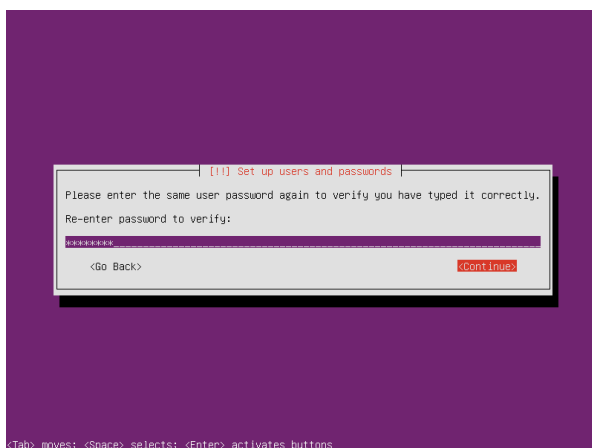
3. UBUNTU

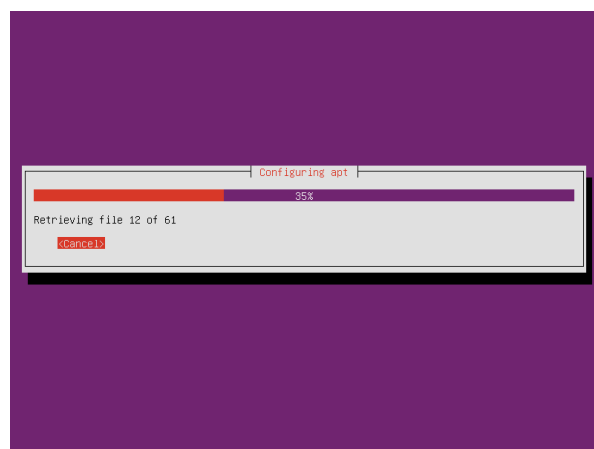
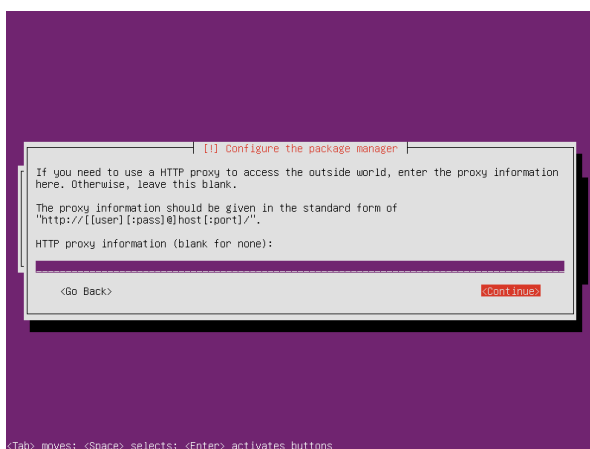
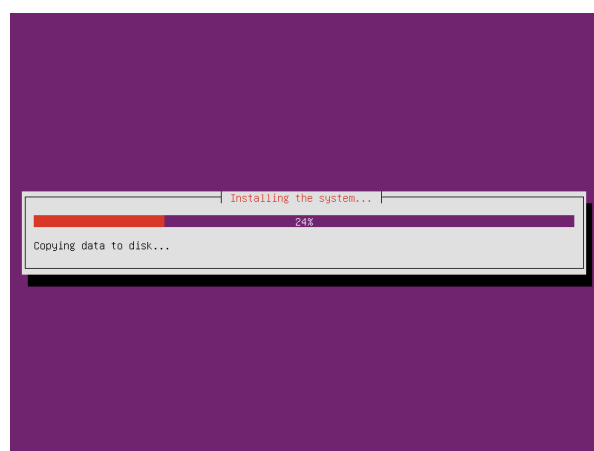
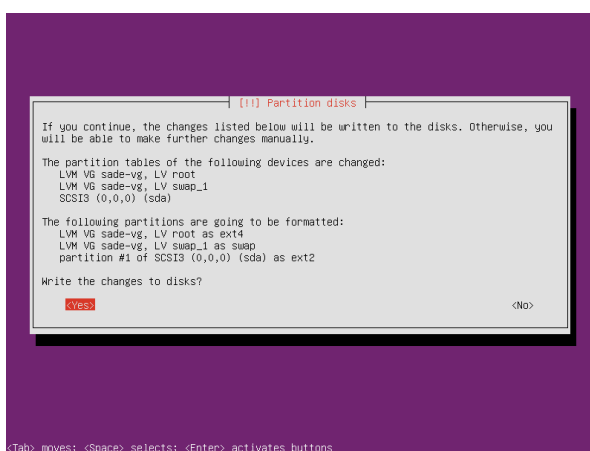
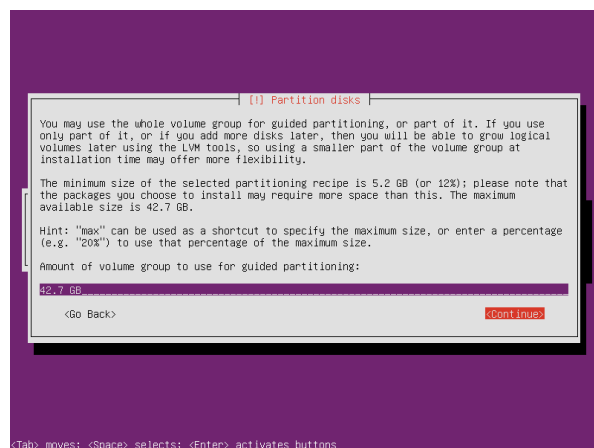
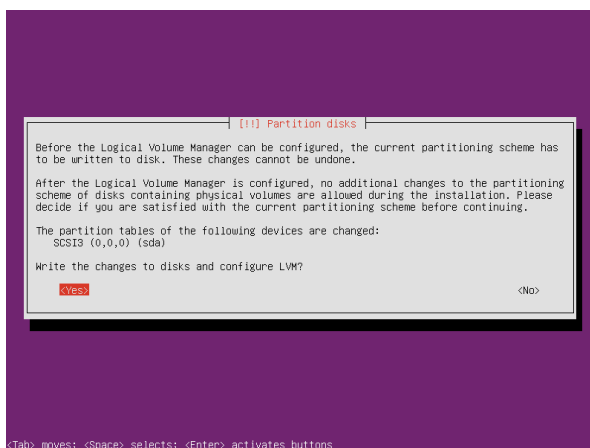
URL	http://releases.ubuntu.com/14.04.1/ubuntu-14.04.1-server-amd64.iso
-----	---

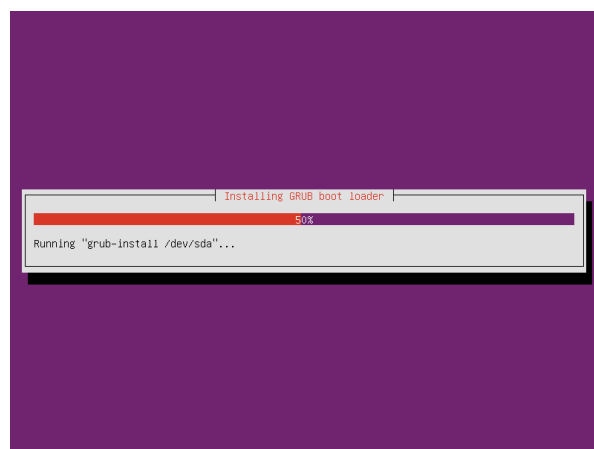
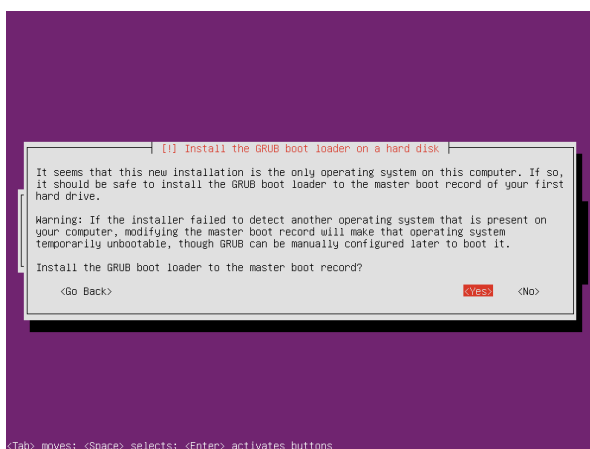
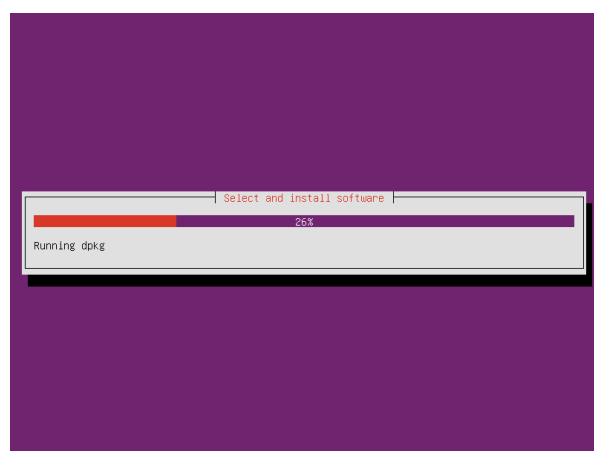
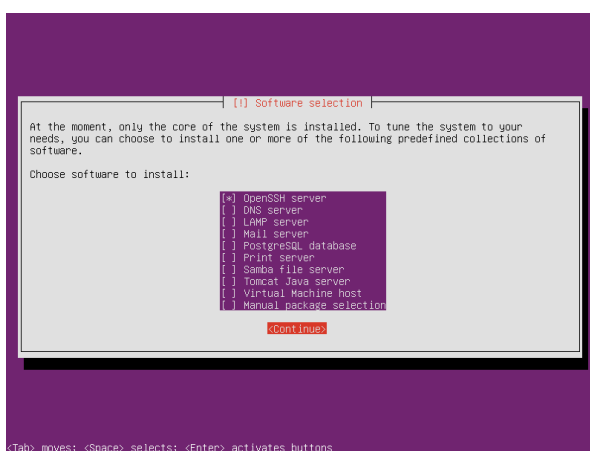
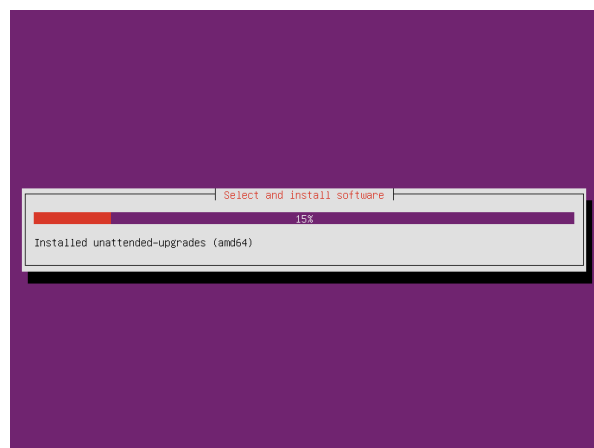
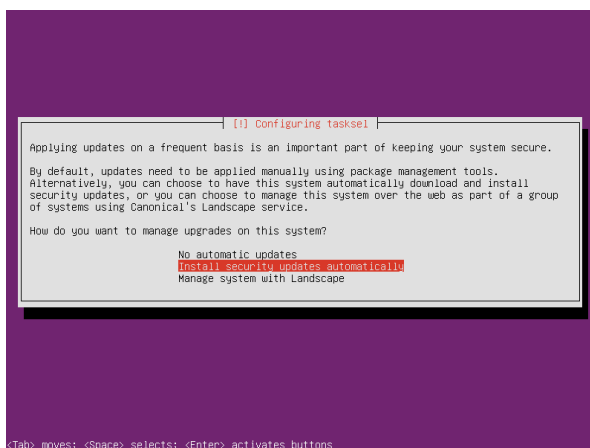
1. Download the ISO image from the above location, write it to a CD and boot a server off this CD.
2. The screen shots below show the steps necessary to do a fresh installation of Ubuntu 14.04 LTS.
3. For documentation purposes, the username and password for the normal user will be:
 john
 password
with the root password also set to the same password. In a real production site, strong passwords must be chosen.
4. After the installation has finished, continue to the next section. From this point on it is assumed that the server is setup and the user logged in. All commands to be typed on the command line is then shown.

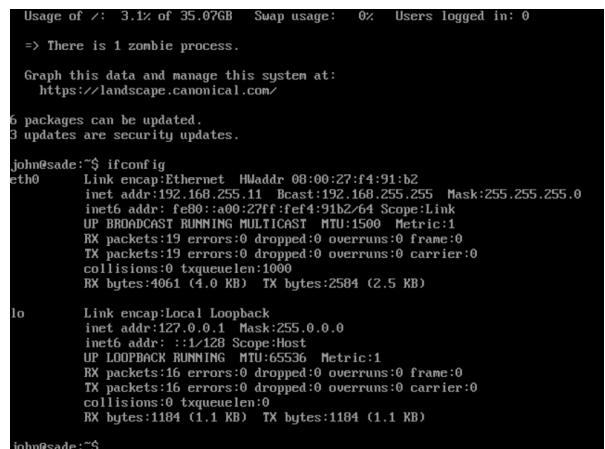
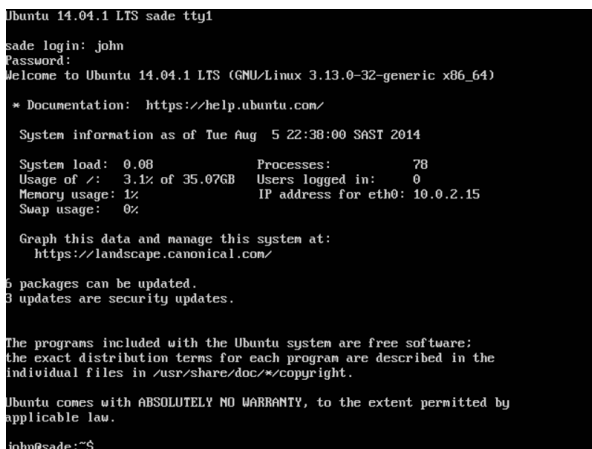
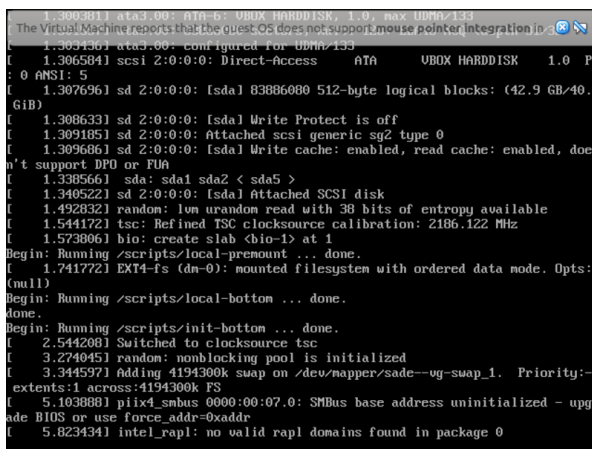
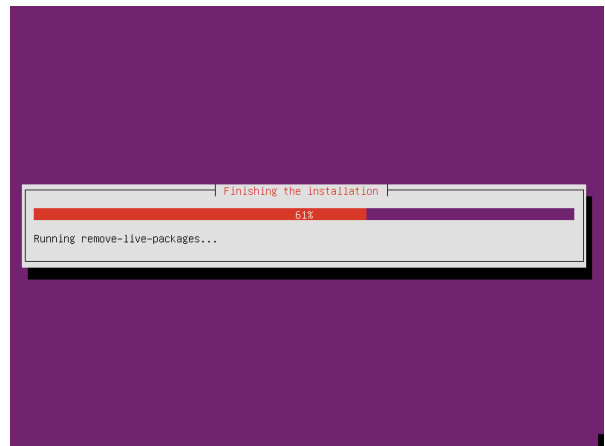
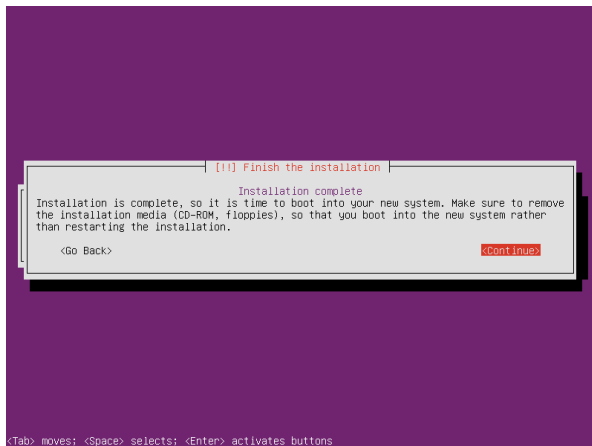












4. UPDATE AND INSTALL PACKAGES

```
$ sudo apt-get update
$ sudo apt-get install git git-core subversion cvs mercurial
$ sudo apt-get install build-essential checkinstall
$ sudo apt-get install ninja gyp libexpat1 libexpat1-dev libfreetype6 libfreetype6-dev
```

5. CONFIGURE GIT

```
$ git config --global user.name "John Smith"
$ git config --global user.email "john@smith.org"
$ git config --global core.autocrlf false
$ git config --global core.filemode false
$ git config --global branch.autosetuprebase always
```

6. CONFIGURE GYP

URL	https://sites.google.com/a/chromium.org/dev/developers/how-tos/api-keys
-----	---

(Follow instructions @ URL to obtain api_key, client_id and client_secret)

```
$ mkdir ~/.gyp
$ vim ~/.gyp/include.gypi
```

```
{
  'variables': {
    'google_api_key': 'your_api_key',
    'google_default_client_id': 'your_client_id',
    'google_default_client_secret': 'your_client_secret',

    # ... other variables you may have ...
  },
}
```

OR (without API keys baked in so it is provided at run-time)

```
$ cat >> ~/.bashrc<<EOD
export GOOGLE_API_KEY='your_api_key'
export GOOGLE_DEFAULT_CLIENT_ID='your_client_id'
export GOOGLE_DEFAULT_CLIENT_SECRET='your_client_secret'
EOD
$ export GOOGLE_API_KEY='your_api_key'
```

```
$ export GOOGLE_DEFAULT_CLIENT_ID='your_client_id'
$ export GOOGLE_DEFAULT_CLIENT_SECRET='your_client_secret'
```

7. INSTALL DEPOT_TOOLS

```
$ sudo mkdir /opt/repos
$ sudo chown john:john /opt/repos
$ cd /opt/repos
$ git clone https://chromium.googlesource.com/chromium/tools/depot_tools.git
$ cat >> ~/.bashrc<<EOD
export PATH=/opt/repos/depot_tools:$PATH
EOD
$ export PATH=/opt/repos/depot_tools:$PATH
```

8. INSTALL JAVA

URL	http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html
-----	---

```
$ mkdir ~/archive
$ cd ~/archive
$ wget --no-cookies --no-check-certificate --header
"Cookie:oraclelicense=accept-securebackup-cookie&gpw_e24=http%3A%2F%2Fwww.oracle.com%2Ftechnetwork%2Fjava%2Fjavase%2Fdownloads%2Fjdk8-downloads-2133151.html"
http://download.oracle.com/otn-pub/java/jdk/7u67-b01/jdk-7u67-linux-x64.tar.gz
$ tar xzvf jdk-7u67-linux-x64.tar.gz
$ sudo mkdir -p /usr/lib/jvm
$ sudo mv jdk1.7.0_67 /usr/lib/jvm

$ cat >> ~/.bashrc<<EOD
export JAVA_HOME=/usr/lib/jvm/jdk1.7.0_67
EOD
$ export JAVA_HOME=/usr/lib/jvm/jdk1.7.0_67

$ sudo update-alternatives --install "/usr/bin/java" "java"
"/usr/lib/jvm/jdk1.7.0_67/bin/java" 1
$ sudo update-alternatives --install "/usr/bin/javac" "javac"
"/usr/lib/jvm/jdk1.7.0_67/bin/javac" 1
$ sudo update-alternatives --install "/usr/bin/javaws" "javaws"
"/usr/lib/jvm/jdk1.7.0_67/bin/javaws" 1
$ sudo update-alternatives --install "/usr/bin/keytool" "keytool"
"/usr/lib/jvm/jdk1.7.0_67/bin/keytool" 1

$ sudo chmod a+x /usr/bin/java
$ sudo chmod a+x /usr/bin/javac
$ sudo chmod a+x /usr/bin/javaws
```

```
$ sudo chmod a+x /usr/bin/keytool
$ sudo chown -R root:root /usr/lib/jvm/jdk1.7.0_67
```

9. COMPILE WEBRTC

```
$ sudo apt-get install libpulse-dev glib-2.0 gtk+-2.0 libglib2.0-doc glib-2.0-dev
gtk+-2.0-dev libnss3-dev libdbus-1-dev libgconf2-dev libgnome-keyring* libudev* g++
libnss3-dev libasound2-dev libjpeg62-dev libxv-dev libgtk2.0-dev libexpat1-dev libasound*
pulseaudio* libxtst-dev libxss-dev libpci-dev
```

```
$ sudo apt-get install apt-file libgll-mesa-glx-lts-trusty g++-multilib gcc-multilib
$ apt-file update
```

```
$ cd /opt/repos
$ mkdir webrtc
$ cd webrtc
$ gclient config http://webrtc.googlecode.com/svn/trunk
```

```
$ gclient sync --force
```

```
$ cd /opt/repos/webrtc/trunk
$ ./build/install-build-deps.sh
```

```
Generate ninja builds
$ cd /opt/repos/webrtc/trunk
$ ninja -C out/Debug
$ ninja -C out/Release
```

10. INSTALL KALDI

```
$ sudo apt-get install libatlas* liblapack* automake autogen autoconf speex libspeex*
```

```
$ cd /opt/repos
$ svn co https://svn.code.sf.net/p/kaldi/code/trunk kaldi
$ svn co -r 4111 https://svn.code.sf.net/p/kaldi/code/sandbox/online kaldi-online
```

```
$ cd /opt/repos/kaldi/tools
$ make
$ ./install_portaudio.sh
$ cd ../src
$ ./configure --shared
NOTE: ignore historic "gcc has a bug in nth_element" warning
$ make depend
$ make
```

```
$ cd /opt/repos/kaldi-online/tools
$ make
$ ./install_portaudio.sh
$ cd ../src
```

```
$ ./configure --shared
$ make depend
$ make
```

11. INSTALL DAHDI

URL	http://downloads.asterisk.org/pub/telephony/dahdi-linux-complete/dahdi-linux-complete-current.tar.gz
-----	---

```
$ tar xzvf dahdi-linux-complete-current.tar.gz
$ cd dahdi-linux-complete-2.9.2+2.9.2
$ make
$ sudo make install
$ sudo make config
```

12. INSTALL LIBPRI

URL	http://downloads.asterisk.org/pub/telephony/libpri/libpri-1.4-current.tar.gz
-----	---

```
$ tar xzvf libpri-1.4-current.tar.gz
$ cd libpri-1.4.15
$ make
$ sudo make install
```

13. INSTALL ASTERISK

URL	http://downloads.asterisk.org/pub/telephony/certified-asterisk/certified-asterisk-1.8.15-current.tar.gz
-----	---

```
$ sudo apt-get install mysql-server mysql-client mysql-workbench mysql-common mysqлтuner
php5-mysql unixODBC unixODBC-dev freetds libmysql* libical* jack libjack-dev resample
libresample1-dev openssl-* libspandsp* sqlite libiksemel* libvorbis* lua5.2 liblua5.2
libgmime-* libsrtp-* net-snmp-* libnewt* libpoppler-dev corosync corosync-dev
libsqlite-dev freetds-dev libcorosync-dev libportaudio2 doxygen graphviz

$ cd ~/archive
```

```

$ wget
http://downloads.asterisk.org/pub/telephony/certified-asterisk/certified-asterisk-1.8.15-
current.tar.gz
$ tar xzvf certified-asterisk-1.8.15-current.tar.gz
$ cd certified-asterisk-1.8.15-current
$ sudo ./configure
$ sudo ./contrib/scripts/get_mp3_source.sh
$ sudo make menuselect
Note: Use default settings (don't need to select anything extra)
$ sudo make
$ sudo make install
$ sudo make progdocs
$ sudo make samples

```

14. INSTALL UNIMRCP-DEPS

URL	http://www.unimrcp.org/project/component-view/unimrcp-deps-1-2-1-tar-gz/download
-----	---

```

$ cd ~/archive
$ wget -O unimrcp-deps-1.2.1.tar.gz http://www.unimrcp.org/project/component-view/unimrcp-deps-1-2-1-tar-gz/download
$ tar xzvf unimrcp-deps-1.2.1.tar.gz
$ cd unimrcp-deps-1.2.1
$ sudo ./build-dep-libs.sh
Note: select all
Note: ignore "[: yes" unexpected operator" error

$ cd libs/apr
$ sudo make
$ sudo make install

$ cd ../apr-util
$ sudo make
$ sudo make install

$ cd ../sofia-sip
$ sudo make
$ sudo make install

```

15. INSTALL SPHINXBASE

URL	http://downloads.sourceforge.net/project/cmusphinx/sphinxbase/0.8/sphinxbase-0.8.ta .gz
-----	---

```

$ sudo mkdir -p /usr/local/share/software/archive

```

```

$ cd /usr/local/share/software/archive
$ sudo wget
http://downloads.sourceforge.net/project/cmusphinx/sphinxbase/0.8/sphinxbase-0.8.tar.gz
$ cd /usr/local/share/software
$ sudo tar xzvf archive/sphinxbase-0.8.tar.gz
$ cd sphinxbase-0.8
$ sudo ./configure
$ sudo make
$ sudo make install

```

16. INSTALL POCKETSPHINX

URL	http://downloads.sourceforge.net/project/cmusphinx/pocketsphinx/0.8/pocketsphinx-0.8.tar.gz
-----	---

```

$ cd /usr/local/share/software/archive
$ sudo wget
http://downloads.sourceforge.net/project/cmusphinx/pocketsphinx/0.8/pocketsphinx-0.8.tar.gz
$ cd /usr/local/share/software
$ sudo tar xzvf archive/pocketsphinx-0.8.tar.gz
$ cd pocketsphinx-0.8
$ sudo ./configure --with-sphinxbase=/usr/local/share/software/sphinxbase-0.8
$ sudo make
$ sudo make install

```

17. INSTALL FLITE

URL	http://www.speech.cs.cmu.edu/flite/packed/flite-1.4/flite-1.4-release.tar.bz2 http://www.speech.cs.cmu.edu/flite/packed/flite-1.4/flite-1.4-WM-8KHzvoices.tar.bz2
-----	--

```

$ cd /usr/local/share/software/archive
$ wget http://www.speech.cs.cmu.edu/flite/packed/flite-1.4/flite-1.4-release.tar.bz2
$ wget
http://www.speech.cs.cmu.edu/flite/packed/flite-1.4/flite-1.4-WM-8KHzvoices.tar.bz2
$ cd /usr/local/share/software
$ sudo tar xjvf archive/flite-1.4-release.tar.bz2
$ cd flite-1.4-release
$ sudo tar xjvf ../archive/flite-1.4-WM-8KHzvoices.tar.bz2
$ sudo ./configure --with-lang --with-vox --with-lex --enable-shared CFLAGS=-fPIC
$ sudo make
$ sudo make install

```


18. INSTALL SADE

```
$ cd /opt/repos
```

For authorised users:

```
$ svn co https://41.181.16.116/svn/repos/sade sade
```

Alternatively, copy from CD

```
$ cp -ar /media/cdrom/SADE-CD/sade-checkout/sade sade
```

19. INSTALL UNIMRCP

```
$ cd /opt/repos
```

```
$ svn checkout https://unimrcp.googlecode.com/svn/trunk unimrcp
```

20. INSTALL UNIMRCP-SOLUTIONS

```
$ cd /opt/repos
```

```
$ svn co https://unimrcp.googlecode.com/svn/solutions unimrcp-solutions
```

21. INSTALL ASTERISK-UNIMRCP

```
$ cd /opt/repos
```

```
$ svn co https://unimrcp.googlecode.com/svn/solutions/asterisk-unimrcp asterisk-unimrcp
```

```
$ cd asterisk-unimrcp
```

```
$ ./bootstrap
```

```
$ ./configure
```

```
$ make
```

```
$ sudo make install
```

22. GENERAL

```
$ sudo vim /etc/ld.so.conf.d/local.conf  
/usr/local/lib
```

```
$ sudo vim /etc/ld.so.conf.d/kaldi.conf  
/opt/repos/kaldi-online/tools/portaudio/install/lib/  
/opt/repos/kaldi-online/tools/openfst/lib  
/opt/repos/kaldi-online/src/lib
```

```

$ sudo vim /etc/profile.d/apr.sh
export PATH=$PATH:/usr/local/apr/bin
$ sudo chmod a+x /etc/profile.d/apr.sh

$ sudo vim /etc/profile.d/java.sh
export JAVA_HOME=/usr/jdk1.7.0_67
export JDK_HOME=/usr/jdk1.7.0_67
export J2RE_HOME=$JAVA_HOME/jre
export PATH=$JAVA_HOME/bin:$J2RE_HOME/bin:$PATH
$ sudo chmod a+x /etc/profile.d/java.sh

$ sudo vim /etc/profile.d/depot_tools.sh
export PATH=/opt/repos/depot_tools:$PATH
$ sudo chmod a+x /etc/profile.d/depot_tools.sh

$ sudo vim /etc/ld.so.conf.d/webrtc.conf
/opt/repos/webrtc/trunk/out/Release
$ sudo /sbin/ldconfig
Note: ignore msg about ./libjingle_peerconnection_so.so not being an ELF file
$ ldd /opt/repos/webrtc/trunk/out/Release/libjingle_peerconnection_so.so

```

23. ADD WEBRTC TO UNIMRCP

Fix /opt/repos/kaldi-online/src/kaldi.mk (for kaldi-online) by commenting out lines to look like this (remove check for dynamic and always add -fPIC to CXXFLAGS):

```

-----
# ifeq ($(KALDI_FLAVOR), dynamic)
# CXXFLAGS += -fPIC
# endif
-----

```

Copy files and directories from /opt/repos/sade/trunk/unimrcp over /opt/repos/unimrcp:

```

$ cp -r /opt/repos/sade/trunk/unimrcp/* /opt/repos/unimrcp/

$ cd /opt/repos/unimrcp
$ ./bootstrap
$ ./configure --disable-flite-plugin --disable-pocketsphinx-plugin
--with-sphinxbase=/usr/local/share/software/sphinxbase-0.8 --enable-kaldi-plugin
--with-kaldi=/opt/repos/kaldi-online --with-apr=/usr/local/apr/bin/apr-1-config
--with-apr-util=/usr/local/apr/bin/apu-1-config --enable-webrtc
--with-webrtc=/opt/repos/webrtc/trunk
$ make
$ sudo make install

```

Edit /usr/local/unimrcp/conf/unimrcpserver.xml and add
 <engine id="Kaldi-1" name="mrcpkaldi" enable="true"/>
 in the 'plugin-factory' section

```

$ cd /usr/local/unimrcp/bin
$ sudo vim start.sh
#!/bin/bash

```

```
export LD_LIBRARY_PATH=/usr/local/apr/lib:/usr/local/unimrcp/lib:$LD_LIBRARY_PATH
cd /usr/local/unimrcp/bin
./unimrcpsvr
$ sudo chmod a+x start.sh
$ ./start.sh
```

24. INSTALLATION-SPECIFIC TASKS

1. Configure Asterisk.
2. Create an IVR script to be copied into /var/lib/asterisk/agi-bin.

- a. For the English SADE municipality hotline demo:

```
ln -s $prompt_dir/sade/eng /opt/repos/csir_nwu/sade/trunk/prompts/sade/cut
cp /opt/repos/csir_nwu/sade/trunk/ivr/sade_recognise.agi \
/var/lib/asterisk/agi-bin/
```

3. Obtain speech recognition models to be placed in /usr/local/unimrcp/data.

- a. For the English SADE municipality hotline demo:

```
svn_dir=/opt/repos/csir_nwu/sade/trunk
ln -s $svn_dir/unimrcp/usr/local/unimrcp/data/nnet_models_2014-07-29
/usr/local/unimrcp/data/nnet_models_2014-07-29
```

4. Configure the UniMRCP server.