INSTITUTO TECNOLÓGICO DE LAS AMÉRICAS

Las Americas Institute of Technology

Asignatura:

Sistemas Operativos III

Tema:

HOWTO Y VIDEO (CENTRAL IP)

Participante:

Santiago M. Duval Contreras

Matricula:

2015-3246

Facilitador:

José Doñe

Fecha:

5/4/2020

HOW-TO? | Instalar un Servidor VOiP en ClearOS

En este documento veremos los pasos requerido para habilitar un servidor de Asterisk en ClearOS - Oracle VirtualBox.

Link a demostración audiovisual: https://youtu.be/QtNtNCvjmZs

Requerimientos del OS:

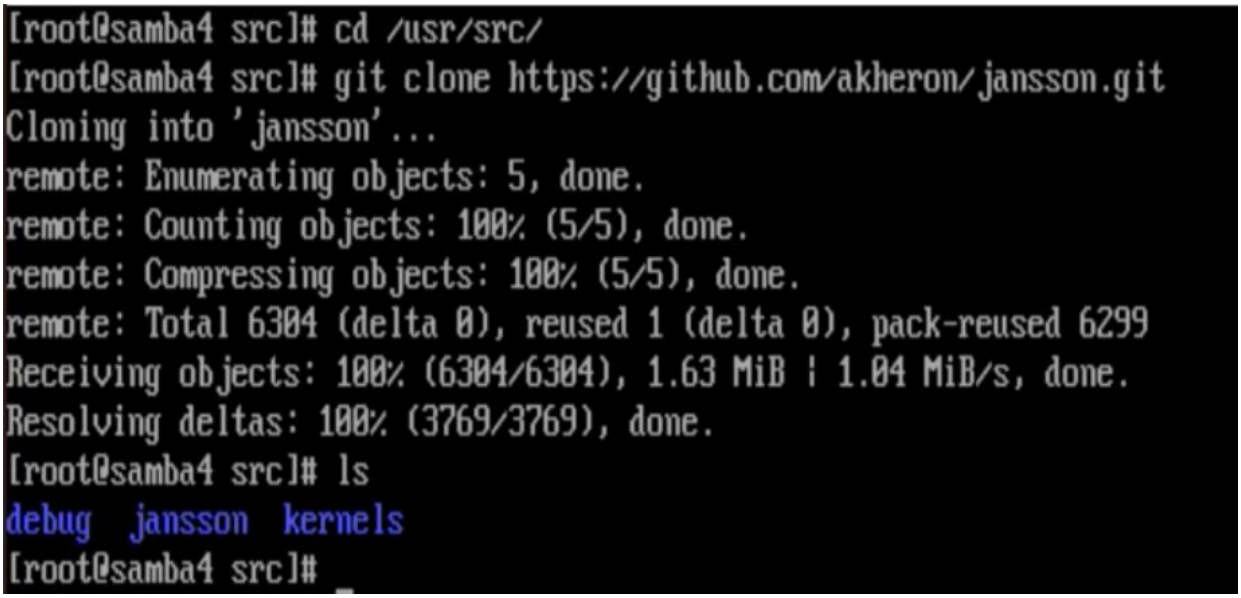
Tener instalado el programa VirtualBox y tener el OS de ClearOS instalado con los requisitos de hardware virtuales que sean requeridos. Selinux tiene que estar desactivado. Todas las maquinas tienen que estar dentro de una misma red interna.

Paso 1 – Instalar los programas requeridos para la compilación de asterisk.

yum -y update
yum groupinstall "Development Tools"
yum install wget net-tools
yum -y install libedit-devel sqlite-devel psmisc gmime-devel ncursesdevel libtermcap-devel sox newt-devel libxml2-devel libtiff-devel
yum -y install audiofile-devel gtk2-devel uuid-devel libtool libuuid subversion kernel-devel kernel-devel-$(uname -r) git subversion
yum -y install kernel-devel crontabs cronie cronie-anacron wget

Paso 2 – Descargar el folder de jannsson.git en el directorio /usr/src/

Usando el comando **cd /usr/src** nos desplazaremos al directorio donde utilizaremos el comando **git clone https://github.com/akheron/jansson.got** clonaremos el folder del jannsson.git requerido para poder compilar Asterisk más adelante.



Paso 3 – Usar el comando cd /jannson para movernos al directorio.

Paso 4 – Preparar los archivos de configuración de jannson.

Utilizando el comando **autoreconf -i** obtendremos los archivos necesarios para compilar Asterisk como por ejemplo el ./configure.

```
[root@samba4 src]# cd jansson/
[root@samba4 jansson]# ls
android       CHANGES       CMakeLists.txt  examples       Makefile.am  scripts
Android.mk    CleanSpec.mk  configure.ac    jansson.pc.in  README.rst   src
appveyor.yml  cmake         doc             LICENSE        release.sh   test
[root@samba4 jansson]# autoreconf -i
libtoolize: putting auxiliary files in AC_CONFIG_AUX_DIR, `.'.
libtoolize: copying file `./ltmain.sh'
libtoolize: Consider adding `AC_CONFIG_MACRO_DIR([m4])' to configure.ac and
libtoolize: rerunning libtoolize, to keep the correct libtool macros in-tree.
libtoolize: Consider adding `-I m4' to ACLOCAL_AMFLAGS in Makefile.am.
configure.ac:13: installing './config.guess'
configure.ac:13: installing './config.sub'
configure.ac:5: installing './install-sh'
configure.ac:5: installing './missing'
parallel-tests: installing './test-driver'
src/Makefile.am: installing './depcomp'
[root@samba4 jansson]# ls
aclocal.m4      CHANGES         config.sub    examples                  ltmain.sh     release.sh
android         CleanSpec.mk    configure     install-sh                Makefile.am   scripts
Android.mk      cmake           configure.ac  jansson.pc.in             Makefile.in   src
appveyor.yml    CMakeLists.txt  depcomp       jansson_private_config.h.in  missing    test
autom4te.cache  config.guess    doc           LICENSE                   README.rst    test-driver
[root@samba4 jansson]#
```

Paso 5 – Configurar jannson.

Para configurar jansson utilizamos el comando **./configure –prefix=/usr/**

```
[root@samba4 jansson]# ./configure --prefix=/usr/
checking for a BSD-compatible install... /bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking for gcc... gcc
```

```
config.status: creating test/suites/Makefile
config.status: creating test/suites/api/Makefile
config.status: creating jansson_private_config.h
config.status: executing depfiles commands
config.status: executing libtool commands
[root@samba4 jansson]#
```

Paso 6 – Compilar e instalar jannson.

Compilamos e instalamos jannson utilizando el comando **make && make install.**

```
make[2]: Entering directory `/usr/src/jansson/test/ossfuzz'
make[3]: Entering directory `/usr/src/jansson/test/ossfuzz'
make[3]: Nothing to be done for `install-exec-am'.
make[3]: Nothing to be done for `install-data-am'.
make[3]: Leaving directory `/usr/src/jansson/test/ossfuzz'
make[2]: Leaving directory `/usr/src/jansson/test/ossfuzz'
make[2]: Entering directory `/usr/src/jansson/test'
make[3]: Entering directory `/usr/src/jansson/test'
make[3]: Nothing to be done for `install-exec-am'.
make[3]: Nothing to be done for `install-data-am'.
make[3]: Leaving directory `/usr/src/jansson/test'
make[2]: Leaving directory `/usr/src/jansson/test'
make[1]: Leaving directory `/usr/src/jansson/test'
make[1]: Entering directory `/usr/src/jansson'
make[2]: Entering directory `/usr/src/jansson'
make[2]: Nothing to be done for `install-exec-am'.
 /bin/mkdir -p '/usr/lib/pkgconfig'
 /bin/install -c -m 644 jansson.pc '/usr/lib/pkgconfig'
make[2]: Leaving directory `/usr/src/jansson'
make[1]: Leaving directory `/usr/src/jansson'
```

Paso 7 – Descargar pjproject con el comando wget.

Utilizamos el comando **cd ..** para volver al directorio usr y utilizamos el comando **wget http://www.pjsip.org/release/2.8/pjproject-2.8.tar.bz2** para descargar los archivos necesarios para compilar Asterisk mas adelante.

```
[root@samba4 jansson]# wget http://www.pjsip.org/release/2.8/pjproject-2.8.tar.bz2
--2020-04-01 11:50:30--  http://www.pjsip.org/release/2.8/pjproject-2.8.tar.bz2
Resolving www.pjsip.org (www.pjsip.org)... 207.38.94.48
Connecting to www.pjsip.org (www.pjsip.org)|207.38.94.48|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://www.pjsip.org/release/2.8/pjproject-2.8.tar.bz2 [following]
--2020-04-01 11:50:35--  https://www.pjsip.org/release/2.8/pjproject-2.8.tar.bz2
Connecting to www.pjsip.org (www.pjsip.org)|207.38.94.48|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/x-bzip2]
Saving to: ‘pjproject-2.8.tar.bz2’

    [         <=>                                     ] 4,976,501   2.35MB/s   in 2.0s

2020-04-01 11:50:38 (2.35 MB/s) - ‘pjproject-2.8.tar.bz2’ saved [4976501]
```

Paso 8 – Descomprimir pjproject-2.8.tar.bz2

Paso 9 – Configurar pjproject.

Utilizando el comando **./configure CFLAGS="-DNDBUG -DPJ_HAS_IPV6=1" –prefix/usr/ --libdir=/usr/lib64 –enable-shared –disable-video –disable-sound –disable-opencore-amr** configuraremos el pjproject que se convertirá en parte del main frame de Asterisk.

```
[root@samba4 pjproject-2.8]# ./configure CFLAGS="-DNDEBUG -DPJ_HAS_IPV6=1" --prefix=/usr --libdir=/u
sr/lib64/ --enable-shared --disable-video --disable-sound --disable-opencore-amr
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking target system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking for g++... g++
checking whether we are using the GNU C++ compiler... yes
checking whether g++ accepts -g... yes
checking for ranlib... ranlib
checking for ar... ar
```

```
config.status: creating pjlib-util/build/os-auto.mak
config.status: creating pjmedia/build/os-auto.mak
config.status: creating pjsip/build/os-auto.mak
config.status: creating third_party/build/os-auto.mak
config.status: creating pjlib/include/pj/compat/os_auto.h
config.status: creating pjlib/include/pj/compat/m_auto.h
config.status: creating pjmedia/include/pjmedia/config_auto.h
config.status: creating pjmedia/include/pjmedia-codec/config_auto.h
config.status: creating pjsip/include/pjsip/sip_autoconf.h


Configurations for current target have been written to 'build.mak', and 'os-auto.mak' in various bui
ld directories, and pjlib/include/pj/compat/os_auto.h.

Further customizations can be put in:
  - 'user.mak'
  - 'pjlib/include/pj/config_site.h'

The next step now is to run 'make dep' and 'make'.

[root@samba4 pjproject-2.8]#
```

Paso 10 – Compilar e Instalar pjproject.

Utilizando el comando **make dep** y **make** compilaremos los archivos necesarios para la instalación de pjproject una vez terminado el proceso del compilado utilizaremos el comando **make install**.

```
gcc -o ../bin/samples/x86_64-unknown-linux-gnu/pjsua2_demo \
     output/pjsua2_sample-x86_64-unknown-linux-gnu/pjsua2_demo.o -L/usr/src/pjproject-2.8/pjlib/lib
-L/usr/src/pjproject-2.8/pjlib-util/lib -L/usr/src/pjproject-2.8/pjnath/lib -L/usr/src/pjproject-2.8
/pjmedia/lib -L/usr/src/pjproject-2.8/pjsip/lib -L/usr/src/pjproject-2.8/third_party/lib          -l
pjsua2 -lstdc++ -lpjsua -lpjsip-ua -lpjsip-simple -lpjsip -lpjmedia-codec -lpjmedia -lpjmedia-videod
ev -lpjmedia-audiodev -lpjmedia -lpjnath -lpjlib-util  -lsrtp -lresample -lgsmcodec -lspeex -lilbcco
dec -lg7221codec -lwebrtc  -lpj -lm -lrt -lpthread
make[2]: Leaving directory `/usr/src/pjproject-2.8/pjsip-apps/build'
make[1]: Leaving directory `/usr/src/pjproject-2.8/pjsip-apps/build'
[root@samba4 pjproject-2.8]#
```

```
mkdir -p /usr/lib64/pkgconfig
sed -e "s!@PREFIX@!/usr!" libpjproject.pc.in | \
       sed -e "s!@INCLUDEDIR@!/usr/include!" | \
       sed -e "s!@LIBDIR@!/usr/lib64!" | \
       sed -e "s/@PJ_VERSION@/2.8/" | \
       sed -e "s!@PJ_LDLIBS@!!" | \
       sed -e "s!@PJ_LDXXLIBS@!-lpjsua2 -lstdc++ -lpjsua -lpjsip-ua -lpjsip-simple -lpjsip -lpjmedi
a-codec -lpjmedia -lpjmedia-videodev -lpjmedia-audiodev -lpjmedia -lpjnath -lpjlib-util  -lsrtp -lre
sample -lgsmcodec -lspeex -lilbccodec -lg7221codec -lwebrtc  -lpj -lm -lrt -lpthread !" | \
       sed -e "s!@PJ_INSTALL_CFLAGS@!!" | \
       sed -e "s!@PJ_INSTALL_CXXFLAGS@!-g -O2 -I/usr/include -DPJ_AUTOCONF=1 -DNDEBUG -DPJ_HAS_IPV6
=1 -DPJ_IS_BIG_ENDIAN=0 -DPJ_IS_LITTLE_ENDIAN=1 -fPIC!" > //usr/lib64/pkgconfig/libpjproject.pc
[root@samba4 pjproject-2.8]# _
```

Paso 11 – Compilar e Instalar pjproject.

Utilizando el comando **cd ..** para movernos al directorio src y con el comando wget procedemos a descargar el source de Asterisk.

```
[root@samba4 pjproject-2.8]# ldconfig
[root@samba4 pjproject-2.8]# cd ..
[root@samba4 src]# wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-16-current.tar
.gz
--2020-04-01 12:02:59--  http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-16-current.ta
r.gz
Resolving downloads.asterisk.org (downloads.asterisk.org)... 76.164.171.238, 2001:470:e0d4::ee
Connecting to downloads.asterisk.org (downloads.asterisk.org)|76.164.171.238|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 27674324 (26M) [application/x-gzip]
Saving to: ‘asterisk-16-current.tar.gz’

100%[=======================================================>] 27,674,324  1.96MB/s   in 41s

2020-04-01 12:03:40 (661 KB/s) - ‘asterisk-16-current.tar.gz’ saved [27674324/27674324]
```

Paso 12 – Nos movemos al directorio Asterisk/contrib/scripts.

Paso 13 – Utilizamos el script ./install_prereq install.

Utilizamos este comando para instalar los últimos programas requeridos para compilar Asterisk. Posteriormente utilizamos el comando yum update para actualizar todos los programas instalados.

```
[root@samba4 asterisk-16.9.0]# cd contrib/scripts/
[root@samba4 scripts]# ls
agents.php              dahdi_span_config_hook   reflocks.py
astcli                  dbsep.cgi                refstats.py
ast_coredumper          file.convert.sh          retrieve_extensions_from_mysql.pl
asterisk.ldap-schema    get_ilbc_source.sh       retrieve_extensions_from_sql.pl
asterisk.ldif           get_mp3_source.sh        retrieve_sip_conf_from_mysql.pl
asterisk.logrotate      get_swagger_ui.sh        safe_asterisk
astgenkey               import-cdr-csv-mysql.pl  safe_asterisk.8
astgenkey.8             install_prereq           safe_asterisk_restart
ast_grab_core           live_ast                 sip_nat_settings
ast_logescalator        loadtest.tcl             sipp-sendfax.xml
ast_loggrabber          lookup.agi               sip_to_pjsip
ast_tls_cert            managerproxy.pl          spandspflow2pcap.log
astversion              messages-expire.pl       spandspflow2pcap.py
autosupport             qview.pl                 valgrind_compare
autosupport.8           README.messages-expire   vmail.cgi
clang-scan-build        refcounter.py            voicemailpwcheck.py
[root@samba4 scripts]# ./install_prereq install
```

```
  urw-base35-gothic-fonts.noarch 0:20170801-10.el7
  urw-base35-nimbus-mono-ps-fonts.noarch 0:20170801-10.el7
  urw-base35-nimbus-roman-fonts.noarch 0:20170801-10.el7
  urw-base35-nimbus-sans-fonts.noarch 0:20170801-10.el7
  urw-base35-p052-fonts.noarch 0:20170801-10.el7
  urw-base35-standard-symbols-ps-fonts.noarch 0:20170801-10.el7
  urw-base35-z003-fonts.noarch 0:20170801-10.el7
  xorg-x11-font-utils.x86_64 1:7.5-21.el7
  xz-devel.x86_64 0:5.2.2-1.el7

Dependency Updated:
  binutils.x86_64 0:2.27-41.base.el7_7.3          curl.x86_64 0:7.29.0-54.el7_7.2
  krb5-libs.x86_64 0:1.15.1-37.el7_7.2            libblkid.x86_64 0:2.23.2-61.el7_7.1
  libcurl.x86_64 0:7.29.0-54.el7_7.2              libmount.x86_64 0:2.23.2-61.el7_7.1
  libsmartcols.x86_64 0:2.23.2-61.el7_7.1         libuuid.x86_64 0:2.23.2-61.el7_7.1
  sqlite.x86_64 0:3.7.17-8.el7_7.1                util-linux.x86_64 0:2.23.2-61.el7_7.1

Complete!
################################################
## install completed successfully
################################################
[root@samba4 scripts]# yum update_
```

Paso 14 – Configuramos Asterisk.

Utilizamos el comando ./configure –libdir=/usr/lib64 –with-jonsson

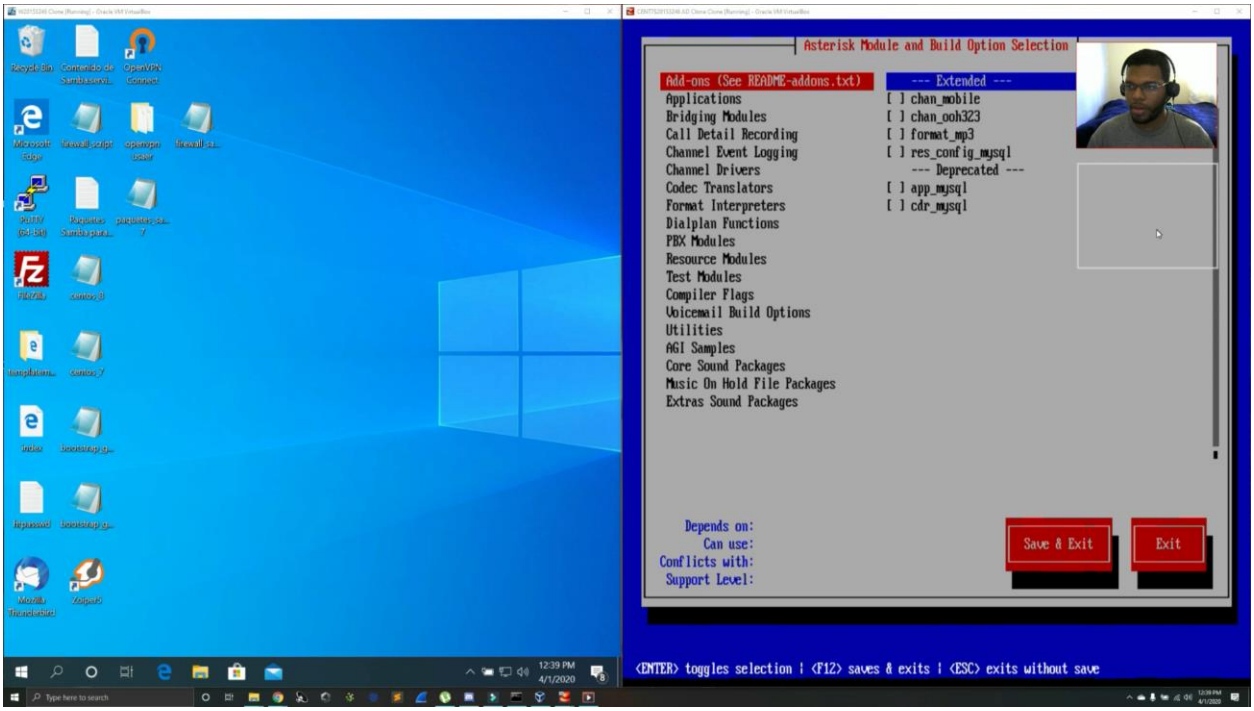
```
[root@samba4 asterisk-16.9.0]# ./configure --libdir=/usr/lib64/
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
```

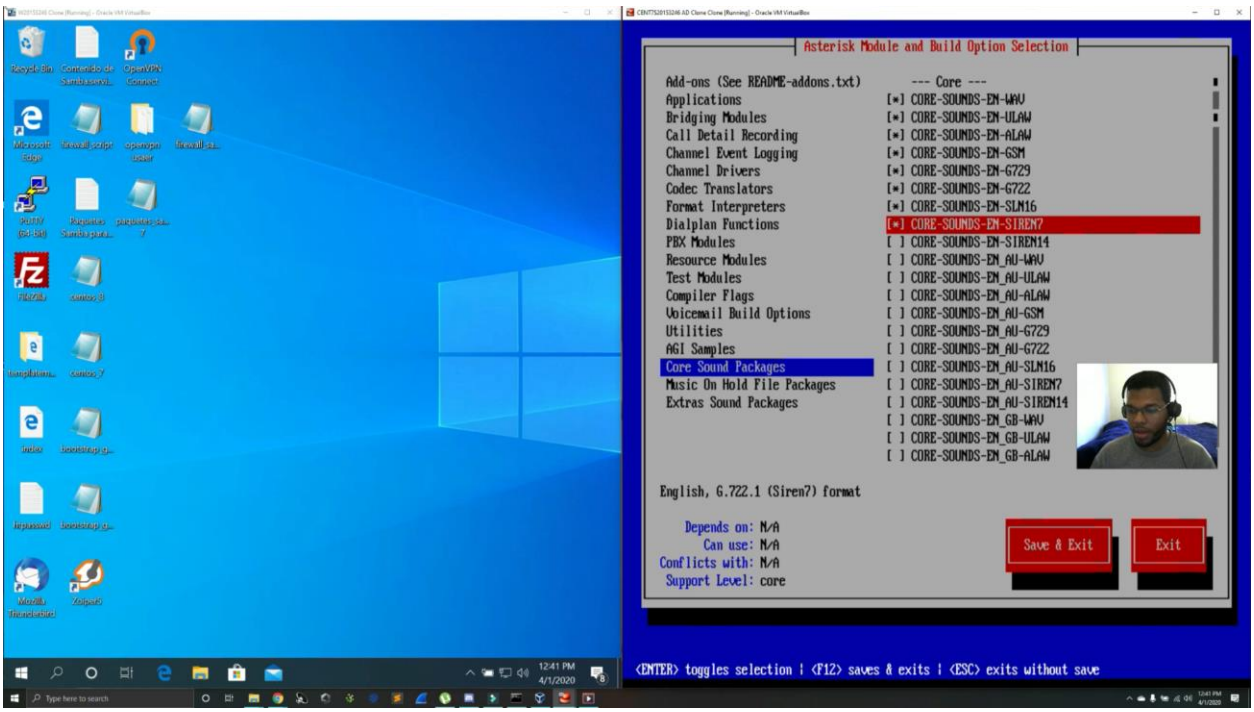
```
             .$$$$$$$$$$$$$$$=..
          .$7$7..          .7$$7:.
        .$$:.                 ,$7.7
      .$7.     7$$$$           .$$77
    ..$$.      $$$$$            .$$$7
   ..7$   .?.  $$$$$   .?.       7$$$.
  $.$.   .$$$7. $$$$7 .7$$$.      .$$$.
 .777.   .$$$$$$77$$$77$$$$$7.     $$$,
 $$$~      .7$$$$$$$$$$$$7.       .$$$.
.$$7         .7$$$$$$$7:          7$$$.
$$$         ??$$$$$$$$$I          .$$$7
$$$       .7$$$$$$$$$$$$$$       :$$$.
$$$       $$$$$$7$$$$$$$$$$$    .$$$.
$$$        $$$   7$$$7  .$$$    .$$$.
$$$$            $$$$7          .$$$.
7$$$7           7$$$$        7$$$
 $$$$$                       $$$
  $$$$7.                     $$  (TM)
   $$$$$$$.           .7$$$$$$  $$
     $$$$$$$$$$$$$7$$$$$$$$$.$$$$$$
       $$$$$$$$$$$$$$$$.

configure: Package configured for:
configure: OS type  : linux-gnu
configure: Host CPU : x86_64
configure: build-cpu:vendor:os: x86_64 : pc : linux-gnu :
configure: host-cpu:vendor:os: x86_64 : pc : linux-gnu :
[root@samba4 asterisk-16.9.0]# _
```

Paso 15 – Seleccionamos que paquetes vamos a instalar con el comando make menuselect.

Ejecutamos el camando make menuselect y procedemos a seleccionar los paquetes que queramos colocar en Asterisk.





Paso 16 – Instalar Asterisk con el comando make install.

```
[root@samba4 asterisk-16.9.0]# make install
   [CC] astcanary.c -> astcanary.o
   [LD] astcanary.o -> astcanary
   [CC] astdb2sqlite3.c -> astdb2sqlite3.o
   [CC] hash/hash.c -> hash/hash.o
   [CC] hash/hash_bigkey.c -> hash/hash_bigkey.o
   [CC] hash/hash_buf.c -> hash/hash_buf.o
   [CC] hash/hash_func.c -> hash/hash_func.o
   [CC] hash/hash_log2.c -> hash/hash_log2.o
   [CC] hash/hash_page.c -> hash/hash_page.o
   [CC] hash/ndbm.c -> hash/ndbm.o
   [CC] btree/bt_close.c -> btree/bt_close.o
   [CC] btree/bt_conv.c -> btree/bt_conv.o
   [CC] btree/bt_debug.c -> btree/bt_debug.o
   [CC] btree/bt_delete.c -> btree/bt_delete.o
   [CC] btree/bt_get.c -> btree/bt_get.o
   [CC] btree/bt_open.c -> btree/bt_open.o
   [CC] btree/bt_overflow.c -> btree/bt_overflow.o
```

```
done
 +---- Asterisk Installation Complete -------+
 +                                           +
 +    YOU MUST READ THE SECURITY DOCUMENT    +
 +                                           +
 + Asterisk has successfully been installed. +
 + If you would like to install the sample   +
 + configuration files (overwriting any      +
 + existing config files), run:              +
 +                                           +
 + For generic reference documentation:      +
 +    make samples                           +
 +                                           +
 + For a sample basic PBX:                   +
 +    make basic-pbx                         +
 +                                           +
 +                                           +
 +-----------------  or ---------------------+
 +                                           +
 + You can go ahead and install the asterisk +
 + program documentation now or later run:   +
 +                                           +
 +               make progdocs               +
 +                                           +
 + **Note** This requires that you have      +
 + doxygen installed on your local system    +
 +-------------------------------------------+
[root@samba4 asterisk-16.9.0]#
```

Paso 17 – Creamos los archivos de configuración de Asterisk.

Crearemos los archivos de configuración de Asterisk utilizando el comando **make samples**, y posteriormente **make config**.

```
/bin/install -c -d "/var/spool/asterisk/voicemail/default/1234/INBOX"
build_tools/make_sample_voicemail "//var/lib/asterisk" "//var/spool/asterisk"
Installing file phoneprov/000000000000.cfg
Installing file phoneprov/000000000000-directory.xml
Installing file phoneprov/000000000000-phone.cfg
Installing file phoneprov/polycom_line.xml
Installing file phoneprov/polycom.xml
Installing file phoneprov/snom-mac.xml
[root@samba4 asterisk-16.9.0]#
```

Paso 18 – Habilitamos el servicio de Asterisk.

Utilizamos el comando **systemctl start asterisk** y **systemctl enable asterisk** para iniciar el proceso de Asterisk y habilitar el servicio para que inicie onboot.

```
● asterisk.service - LSB: Asterisk PBX
   Loaded: loaded (/etc/rc.d/init.d/asterisk; bad; vendor preset: disabled)
   Active: active (running) since Wed 2020-04-01 14:21:01 EDT; 23s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 26841 ExecStart=/etc/rc.d/init.d/asterisk start (code=exited, status=0/SUCCESS)
 Main PID: 26871 (asterisk)
   CGroup: /system.slice/asterisk.service
           ├─26868 /bin/sh /usr/sbin/safe_asterisk
           └─26871 /usr/sbin/asterisk -f -vvvg -c
```

Paso 19 – Nos movemos con el comando cd al directorio /etc/asterisk.

Paso 20 – hacemos una copia del archivo sip.conf

Utilizamos el comando nano sip.conf y guardamos el archivo sip.conf como sip.conf.example, después esto eliminamos el archivo sip.conf.

```
asterisk.adsi           config_test.conf        musiconhold.conf        sip.conf
asterisk.conf           console.conf            muted.conf              sip.conf.example
```

```
cel.conf                hep.conf                res_corosync.conf       vpb.conf
cel_custom.conf         http.conf               res_curl.conf           xmpp.conf
cel_odbc.conf           iax.conf                res_fax.conf
[root@samba4 asterisk]# rm sip.conf
sip.conf          sip.conf.example
[root@samba4 asterisk]# rm sip.conf
rm: remove regular file ‘sip.conf’? y
[root@samba4 asterisk]#
```

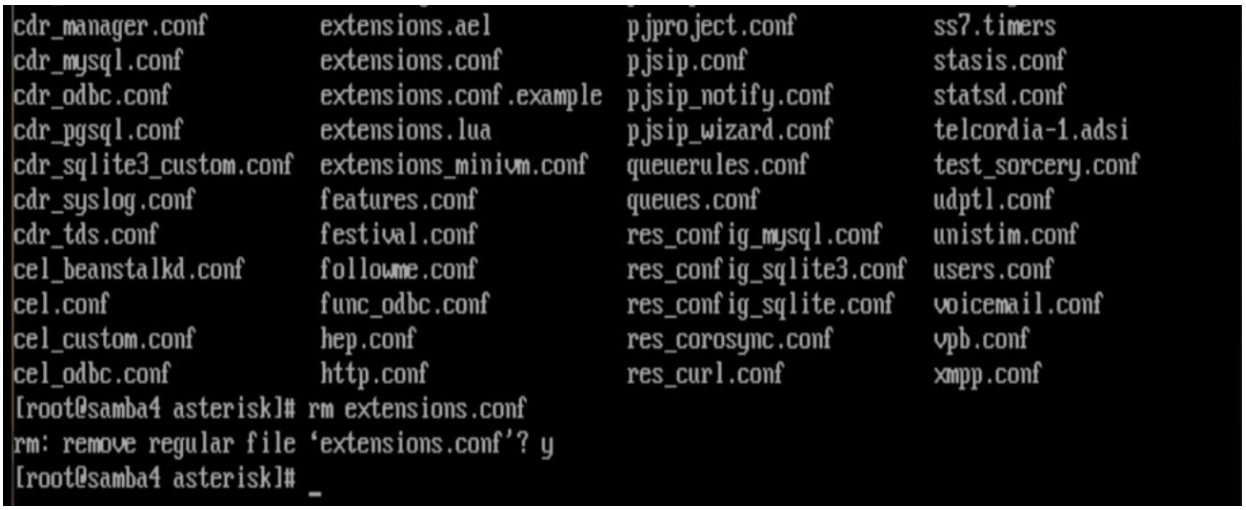
Paso 21 – Creamos un nuevo archivo sip.conf

Utilizamos el comando nano sip.conf para crear un nuevo sip.conf y añadimos la siguiente información:

[default]
port=5060
[ast](¡)
type= friend
context= default
host= dynamic
[numero de la extensión]
callerid = nombre de la cuenta
secret= password de la cuenta
[numero de la extensión]
callerid = nombre de la cuenta
secret= password de la cuenta

Paso 22 – hacemos una copia del archivo extensions.conf

Utilizamos el comando nano extensions.conf y guardamos el archivo extensions.conf como extensions.conf.example, después esto eliminamos el archivo extensions.conf.



Paso 23 – Creamos un nuevo archivo extensions.conf

Utilizamos el comando nano extensions.conf para crear un nuevo extensions.conf y añadimos la siguiente información:

[default]
Exten=> _XXX,1,Dial(SIP/${EXTEN})
Same=> n,Hangup(16)

Paso 24 – Ejecutamos los siguientes comandos para crear el grupo de asterisk y cambiar el propietario de los archivos de Asterisk.

- **groupadd asterisk**
- **useradd -r -d /var/lib/asterisk/ -g asterisk asterisk**
- **usermod -aG audio,dialout asterisk**
- **chown -R asterisk.asterik /etc/asterisk /var/{lib,log,spool}/asterisk /usr/lib64/Asterisk**

Paso 25 – Editar las siguientes líneas del archivo Asterisk

Utilizamos el comando nano /etc/sysconfig/asterisk y eliminaremos el comentario de las líneas:

#AST_USER="asterisk"

#AST_GROUP="asterisk"

Paso 26 – Editar las siguientes líneas del archivo asterisk.conf

Utilizamos el comando nano /etc/asterisk/asterisk.conf y añadiremos estas líneas:

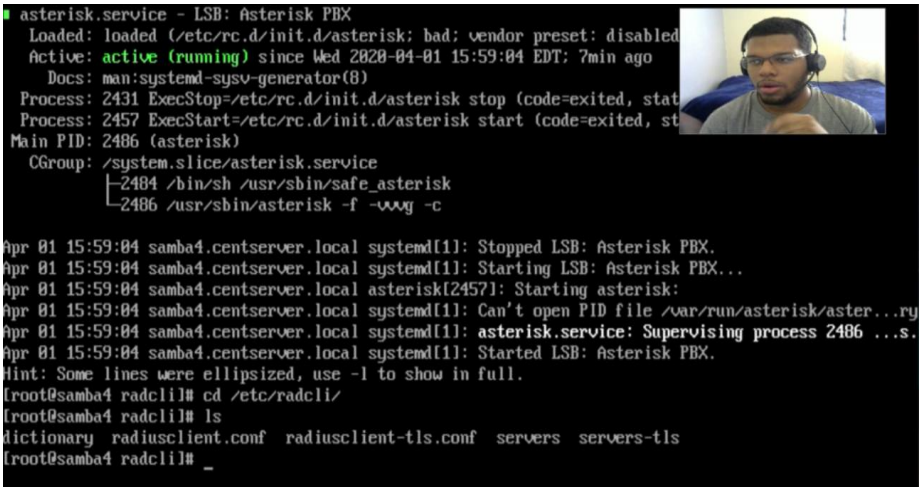
runuser= asterisk ;The user to run as.

rungroup= Asterisk ;The group to run as.

Paso 27 – Creamos el directorio radiusclient-ng en el directorio /etc/

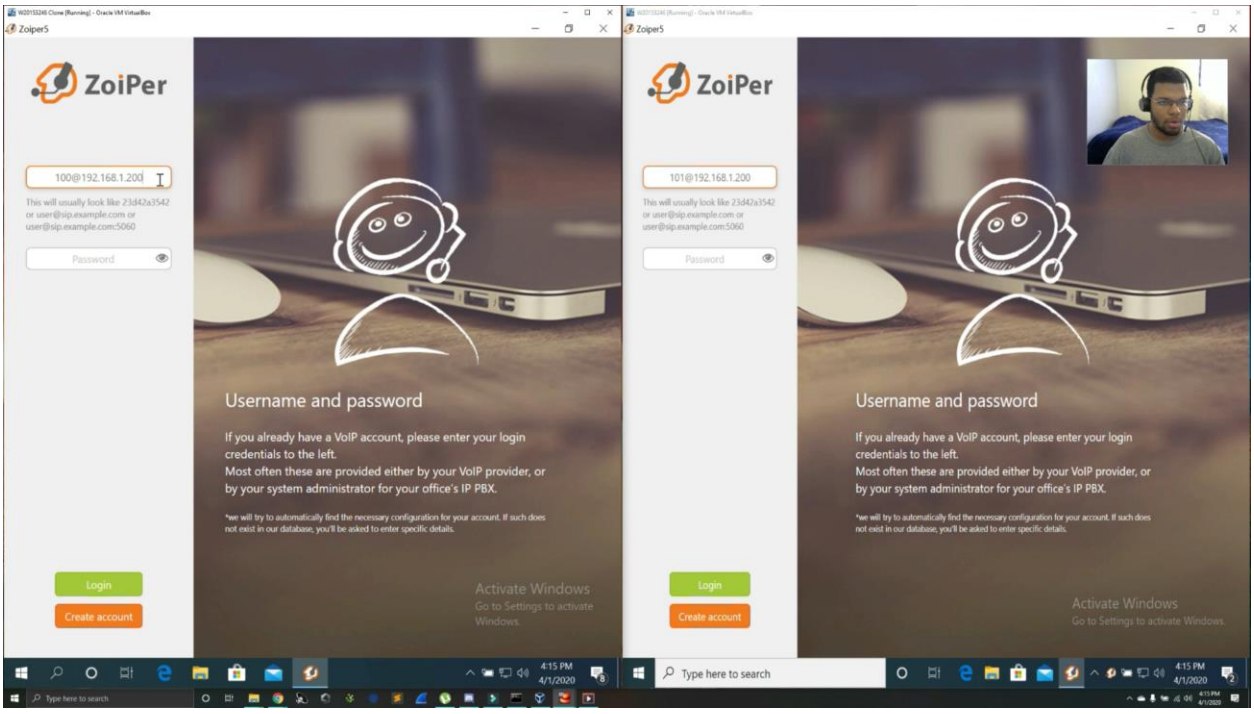
Paso 28 – Movemos el archivo radiusclient.conf desde /etc/radcli a /etc/radiusclient-ng

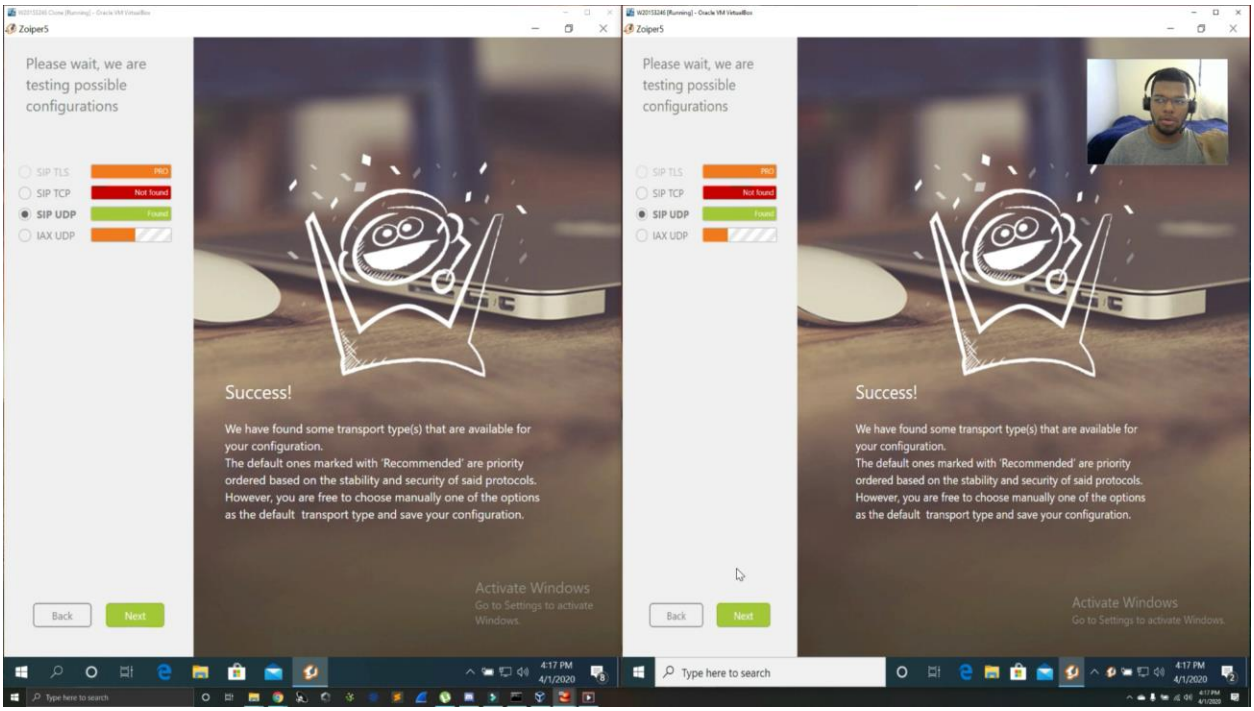
Paso 29 – Reiniciar el servicio de Asterisk.

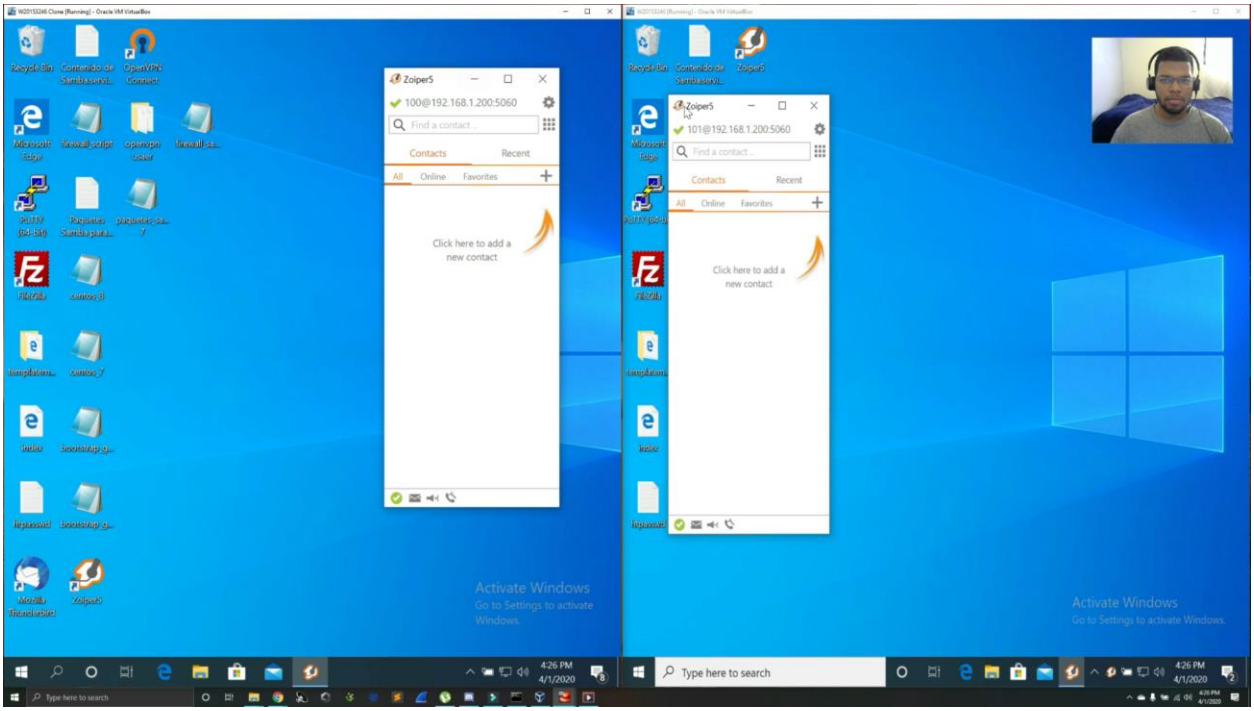


Paso 30 – Abrir el servicio de Asterisk con el comando Asterisk -rv.

Paso 31 –Añadir las cuentas de Asterisk a zoiper 5.




Zoiper5
Please wait, we are testing possible configurations
SIP TLS
SIP TCP
SIP UDP
IAX UDP
Success!
We have found some transport type(s) that are available for your configuration.
The default ones marked with 'Recommended' are priority ordered based on the stability and security of said protocols.
However, you are free to choose manually one of the options as the default transport type and save your configuration.
Back
Next
Activate Windows
Go to Settings to activate Windows.
Type here to search


Zoiper5
100@192.168.1.200:5060
101@192.168.1.200:5060
Find a contact ..
Contacts
Recent
All
Online
Favorites
Click here to add a new contact
Activate Windows
Go to Settings to activate Windows.
Type here to search

Paso 32 – Probando el VOiP entre ambos clientes.