CTEngine Installation

Technical procedure

VERSION 1.0 (2020.11.15)

Restricted use

The following installation example has been made and tested on Debian 9 (x64) minimal.

However, all following components can be installed also on other Linux distributions, Windows or other operating systems.

Table of Contents

1 Asterisk

Supported Asterisk versions: min. 12, currently max. 16.14. Versions 12, 14 and 15 are not recommended because they are not LTS. Highly recommended version is 16. In the following procedure we are using version 16.13.

1.1 Download

Download asterisk from https://downloads.asterisk.org/pub/telephony/asterisk/

apt-get update
apt-get install wget –y
cd /usr/local/src/ && wget
https://downloads.asterisk.org/pub/telephony/asterisk/asterisk-16.13.0.tar.gz
tar xfzv asterisk-16.13.0.tar.gz && cd asterisk-16.13.0/

1.2 DAHDI

If you plan to use TDM (ISDN) - Digium/Sangoma digital cards, make sure you install the following:

- DAHDI
- LIBPRI

Check the following documentation on how to do that:
https://wiki.asterisk.org/wiki/pages/viewpage.action?pageId=4817506
https://wiki.asterisk.org/wiki/display/AST/Building+and+Installing+DAHDI
https://wiki.asterisk.org/wiki/display/AST/Building+and+Installing+LibPRI

Otherwise, skip this step and go to the next one.

1.3 Checking Requirements

This might take a while.

cd contrib/scripts && ./install_prereq install –y
cd ../../ && ./configure --with-jansson-bundled

1.4 Build and install Asterisk

Command “make” might take a while.

make menuselect (optional)
Core Sound Packages - select EN-WAV, EN-ULAW, EN-ALAW, EN-GSM
Music On Hold File Packages - select WAV, ULAW, ALAW, GSM
Extras Sound Packages - select EN-WAV, EN-ULAW, EN-ALAW, EN-GSM
Save & Exit

make
make install

1.5 Install sample files

make samples

1.6 Install init/startup script and logrotate

make config && make install-logrotate

1.7 Verify status

systemctl enable asterisk
systemctl start asterisk
systemctl status asterisk

If you’ve made step 2, check for DAHDI as well:
https://wiki.asterisk.org/wiki/display/AST/Validating+Your+Installation

1.8 Configure modules

This can be made manually or via FreePBX GUI tool which is not described in this document and must be examined by your own. More information about it can be found on https://www.freepbx.org/. Manual configuration means editing Asterisk configuration files in directory /etc/asterisk:

- backup all files from the directory
- remove all unnecessary .conf files in order to have minimal configuration. Usually we keep only following files in the directory:

- ari.conf
- ast_debug_tools.conf
- asterisk.conf
- cdr.conf
- cdr_custom.conf
- chan_dahdi.conf (only if TDM is used)
- cli.conf
- cli_aliases.conf
- cli_permissions.conf
- dahdi-channels.conf (only if TDM is used)
- extensions.conf
- http.conf
- iax.conf
- logger.conf
- modules.conf
- pjproject.conf
- pjsip.conf
- pjsip_notify.conf
- pjsip_wizard.conf
- rtp.conf
- sip.conf (deprecated in newer versions of Asterisk, PJSIP recommended)
- sorcery.conf
- ss7.timers
- stasis.conf

Of course, if you need any other configuration file for modules you are using, be free to keep it in the directory.

How to configure all of this files is not a scope of this document except configuration files that are necessary for CTEngine be able to communicate with Asterisk which is covered in chapter Connect Asterisk and CTEngine.

However, examples of all this .config files can be found and downloaded from our CTEngine website/platform.

2 Java

Minimal supported Java version is 8. Newer versions are also possible, but not properly tested until now.

2.1 Download

Download Java 8 from from Java.com / Oracle website:

```
# cd /opt/ && wget --no-cookies --no-check-certificate \
--header "Cookie: oraclelicense=accept-securebackup-cookie" \
https://javadl.oracle.com/webapps/download/AutoDL?BundleId=242980_a4634525489241b9a9e1aa73d9e118e6 -O jre-8u261-linux-x64.tar.gz

# tar xfzv jre-8u261-linux-x64.tar.gz
```

3 CTEngine

To be able to download and install CTEngine you have to be registered user with appropriate download license.

3.1 Download

Login to the CTEngine website/platform and download the latest compressed ctengine.zip file. Place the file into your directory and extract it.

3.2 Init / start script

Check README file in order to configure init/startup script which is used to start CTEngine software on system boot.

3.3 Install certificates

Optionally you can add SSL certificates into directory CTENGINE_HOME/conf/cert.

3.4 Configuration

Configure the main configuration file CTENGINE_HOME/ct_engine.cfg as follows:

```
## CT ENGINE CONFIGURATION

# Log4J logging
# Pattern desc: c - category, C - class, d - date (%d{dd MMM yyyy HH:mm:ss,SSS}), F -
file, L - line,
# l - source location, m - message, M - method, n - \n, p - priority, r - ms from start, t
- thread
# Set root logger level to DEBUG and two appenders

log4j.rootLogger=TRACE, stout, logfile
log4j.logger.ch.loway=INFO
log4j.logger.io.netty=INFO
#log4j.additivity.CDR=false
log4j.additivity.CDR2=false

# logger used for logging CDRs
#log4j.logger.CDR=INFO, cdrfile
log4j.logger.CDR2=INFO, cdrfile2

# 'stout' is set to be a ConsoleAppender.
log4j.appender.stout=org.apache.log4j.ConsoleAppender
log4j.appender.stout.layout=org.apache.log4j.PatternLayout
log4j.appender.stout.layout.ConversionPattern=%d{yy-MM-dd HH:mm:ss.SSS} %-5p | %-25.25t |
%c{1} - %m%n

# 'logfile' is set to be FileAppender that rolls every day
log4j.appender.logfile=org.apache.log4j.DailyRollingFileAppender
log4j.appender.logfile.Append=true
log4j.appender.logfile.File=/var/log/ct_engine/ct_engine.log
log4j.appender.logfile.DatePattern='-'yyyy-MM-dd'.log'
log4j.appender.logfile.layout=org.apache.log4j.PatternLayout
log4j.appender.logfile.layout.ConversionPattern=%d{yy-MM-dd HH:mm:ss.SSS} %-5p | %25.25t |
%20.20c{1} - %m%n

# 'cdrfile' is set to be FileAppender that rolls every day
#log4j.appender.cdrfile=org.apache.log4j.DailyRollingFileAppender
#log4j.appender.cdrfile.Append=true
#log4j.appender.cdrfile.File=/usr/local/ct_engine/cdrs/cdr.log
#log4j.appender.cdrfile.DatePattern='-'yyyy-MM-dd'.log'
#log4j.appender.cdrfile.layout=org.apache.log4j.PatternLayout
#log4j.appender.cdrfile.layout.ConversionPattern=%d{yy-MM-dd HH:mm:ss.SSS} | %m%n

# 'cdrfile2' is set to be FileAppender that rolls every day
log4j.appender.cdrfile2=org.apache.log4j.DailyRollingFileAppender
log4j.appender.cdrfile2.Append=true
log4j.appender.cdrfile2.File=/usr/local/ct_engine/cdrs/cdr2.log
log4j.appender.cdrfile2.DatePattern='-'yyyy-MM-dd'.log'
log4j.appender.cdrfile2.layout=org.apache.log4j.PatternLayout
log4j.appender.cdrfile2.layout.ConversionPattern=%d{yy-MM-dd HH:mm:ss.SSS} | %m%n

##### Engine settings #####

engine.name=CTEngine Dev
engine.config-dir=conf
engine.apps.root-dir=apps
engine.plugins.root-dir=plugins

##### HTTP Server settings #####
#http-server.host=10.0.10.1
http-server.host=ctengine.yourdomain.com
http-server.port=8090
http-server.ssl=true
http-server.ssl.cert-chain=fullchain.pem
http-server.ssl.priv-key=privkey.pem
http-server.webroot=webroot

##### SMTP Server settings #####
smtp-server.host=mail.yourdomain.com
smtp-server.port=25
smtp-server.from=ctengine@yourdomain.hr

#### Languages ####
language.1=en:English
language.2=en-us:English US
language.3=hr:Hrvatski
language.4=hu:Magyar

##### CDR settings #####
cdr.print.conference.entered-left=false

##### Window settings #####
# window: false (default) / true
window.show=false
```

4 Connect Asterisk and CTEngine

In order to control Asterisk, CTEngine has to connect to its ARI interface.

4.1 ARI

Edit ARI configuration file /etc/asterisk/ari.conf as follows:

```
[general]
enabled = yes
pretty = no
allowed_origins = 127.0.0.1,http://ari.asterisk.org
;auth_realm =
;websocket_write_timeout = 100
;channelvars = var1,var2,var3

[ariUsername]
type = user
read_only = no
password = ariPassword
password_format = plain
```

This is just an example of ari.conf file, but you are responsible to set your own secret values in all of this fields. The most important parameters you HAVE to change:

- allowed_origins: set it only to the IP address or domain from where CTEngine will connect to it
- ari-username: change it to your own secret username that will be used by CTEngine
- ari-password: change it to your own secret password that will be used by CTEngine
- password_format may be set to plain (the default) or crypt. When set to crypt, crypt(3) is used to validate the password. A crypted password can be generated using mkpasswd -m sha-512

4.2 Extensions

Edit extensions configuration file /etc/asterisk/extensions.conf as follows:

```
[general]
static=yes
writeprotect=no
clearglobalvars=no
[globals]
CONSOLE=Console/dsp
TRUNKMSD=1

[from-pri]
exten =>
_X!,1,Stasis(AriAppName,SPAN=${CHANNEL(dahdi_span)},CHANNEL=${CHANNEL(dahdi_channel)},GR
OUP=${GROUP_NR},TYPE=${CHANNEL(dahdi_type)},B-CHANNEL=${CHANNEL(no_media_path)},CALLER-
ANI=${CALLERID(ani)},CALLER-ANI2=${CALLERID(ani2)},CALLER-NUM=${CALLERID(num)},CALLER-
NUM-VALID=${CALLERID(num-valid)},CALLER-NAME=${CALLERID(name)},CALLER-NAME-
VALID=${CALLERID(name-valid)},CALLER-SUBADDR=${CALLERID(subaddr)},CALLER-SUBADDR-
VALID=${CALLERID(subaddr-valid)},CALLED-NUM=${CALLERID(dnid)},CALLED-
SUBADDR=${CALLERID(dnid-subaddr)},CALLED-SUBADDR-VALID=${CALLERID(dnid-subaddr-
valid)},AUDIO-NATIVE=${CHANNEL(audionativeformat)})

[from-pjsip]
exten => _.,1,Stasis(AriAppName,TRUNK=${CHANNEL(endpoint)},CALLER-
ALL=${CALLERID(all)},CALLER-NUM=${CALLERID(num)},CALLER-NAME=${CALLERID(name)},CALLER-
NAME-VALID=${CALLERID(name-valid)},CALLED-NUM=${EXTEN},SIP-
FROM=${PJSIP_HEADER(read,From)},SIP-TO=${PJSIP_HEADER(read,To)},SIP-
CONTACT=${PJSIP_HEADER(read,Contact)},SIP-
VIA=${PJSIP_HEADER(read,Via)},USERAGENT=${PJSIP_HEADER(read,User-Agent)},AUDIO-
NATIVE=${CHANNEL(audionativeformat)},AUDIO-READ=${CHANNEL(audioreadformat)},AUDIO-
WRITE=${CHANNEL(audiowriteformat)},VIDEO-NATIVE=${CHANNEL(videonativeformat)})

[from-iax2]
exten => _.,1,Stasis(AriAppName,TRUNK=${CHANNEL(peername)},CALLER-
ALL=${CALLERID(all)},CALLER-NUM=${CALLERID(num)},CALLER-NAME=${CALLERID(name)},CALLER-
NAME-VALID=${CALLERID(name-valid)},CALLER-SUBADDR=${IAXVAR(CALLER-SUBADDR)},CALLER-
SUBADDR-VALID=${IAXVAR(CALLER-SUBADDR-VALID)},CALLED-NUM=${EXTEN},CALLED-
SUBADDR=${IAXVAR(CALLED-SUBADDR)},CALLED-SUBADDR-VALID=${IAXVAR(CALLED-SUBADDR-
VALID)},PEERIP=${CHANNEL(peerip)},AUDIO-NATIVE=${CHANNEL(audionativeformat)},AUDIO-
READ=${CHANNEL(audioreadformat)},AUDIO-WRITE=${CHANNEL(audiowriteformat)},VIDEO-
NATIVE=${CHANNEL(videonativeformat)})

[from-sip]
exten => _.,1,Stasis(AriAppName,TRUNK=${CHANNEL(peername)},CALLER-
ALL=${CALLERID(all)},CALLER-NUM=${CALLERID(num)},CALLER-NAME=${CALLERID(name)},CALLER-
NAME-VALID=${CALLERID(name-valid)},CALLED-NUM=${EXTEN},SIP-FROM=${SIP_HEADER(From)},SIP-
TO=${SIP_HEADER(To)},SIP-
CONTACT=${SIP_HEADER(Contact)},PEERIP=${CHANNEL(peerip)},RECVIP=${CHANNEL(recvip)},RECVP
ORT=${CHANNEL(recvport)},USERAGENT=${CHANNEL(useragent)},RTP-SRC-
AUDIO=${CHANNEL(rtpsource,audio)},RTP-SRC-VIDEO=${CHANNEL(rtpsource,video)},RTP-DEST-
AUDIO=${CHANNEL(rtpdest,audio)},RTP-DEST-VIDEO=${CHANNEL(rtpdest,video)},AUDIO-
NATIVE=${CHANNEL(audionativeformat)},AUDIO-READ=${CHANNEL(audioreadformat)},AUDIO-
WRITE=${CHANNEL(audiowriteformat)},VIDEO-NATIVE=${CHANNEL(videonativeformat)})
```

This is the main part of the configuration that is usually just enough to connect Asterisk and CTEngine. **AriAppName** should be replaced with your value which will be also set up in CTEngine configuration file CTENGINE_HOME/conf/servers.cfg.

There are other possibilities that could be included into extensions.conf file such as reading out caller/callee subadresses from SIP protocol and forwarding it into CTEngine ARI application, but for this options please contact our support.

4.3 CTEngine servers

Edit CTEngine configuration file CTENGINE_HOME/conf/servers.cfg:

```
# CTEngine servers configuration file
@version 1.0

[server]
id=1
name=CTENGINE-NAME
auto-start=true
selected=true

channels.tdm=30
channels.ip=30
channels.sw=10
channels.in.distribution=ROUND_ROBIN
channels.in.reservation.tdm=ALL
channels.in.reservation.ip=ALL
channels.out.tdm.groups=1:1-10;2:11-20;2:21-30
channels.app.tdm.1=ALL;hr.maxcom.ct.app.router.RouteStart;true
channels.app.ip.1=ALL;hr.maxcom.ct.app.router.RouteStart;true

driver.class=hr.maxcom.ct.driver.asterisk_ari.AsteriskAriCTDriver
driver.auto-start=true

# Driver Asterisk ARI specific settings
driver.ari.application=AriAppName
driver.ari.host=127.0.0.1
driver.ari.port=8088
driver.ari.username=ariUsername
driver.ari.password=ariPassword
driver.ari.version=4.1.3
```

If you are not using TDM board you can remove or comment out all parameters containing word ‘tdm’.

Parameters starting with ‘driver.ari.’ have to match to previously configured parameter values set in Asterisk configuration, otherwise CTEngine will fail to connect to it .

‘driver.ari.port’ is the port configured in Asterisk configuration file http.conf, parameter ‘bindport’.

‘driver.ari.version’ depends of the Asterisk version you are using. More details can be found on web site https://github.com/l3nz/ari4java/wiki/Asterisk-Version-to-ARI-Version