ODIN project

Monitoring system for Asterisk IPBX. This project is has 4 parts

Http Server: Web server for monitoring iPBX states by web interface. This application is base inot nodejs.

OdinAMI : Dispatcher Asterisk AMI Events to the different types of client. REDIS message queue used as message dispatcher.

OdinF1COM : F1COM server some kind of glue beetwen OdinAMI and F1COM clients

FastAGI : There are two FastAGI servers for incomming call and for call monitoring.

Web Server

Nodejs web server based. Client part of the application implemented with angularsjs.

OdinAMI

Python twisted server based on starpy. Monitor the states of an Asterisk IPBX to different type of clent and accept(dispatch) request to Asterisk IPBX.

OdinF1COM

Python twisted server. Its role is to manage F1COM protocol for outgoing and incomming calls. Configuration file contains values for different parameters. Theses parameters are divided into several sections. Section names and settings are very meaningful to use. Please see the configuration file odinf1com.conf into config directory

Configuration of Asterisk dialplan is very important for the application. Please pay a littele attention for the dialplan configuration. Dialplan example is presented below:

```
[incomming-calls-for-queue]
exten = _X.,1,NoOp(incomming-calls-for-queue)
same => n,Answer
same => n,UserEvent(incommingcall,Context:from-white-house, channel: ${CHANNEL},
```

```
extention:${EXTEN},calleridnum:${CALLERID(num)},calleridname:${CALLERID(name)},uniqueid: ${(
same => n,Goto(queues,6500,1)
same => n,Hangup()
```

Generated UserEvent is important for the incomming call processing don't forget to include the context to yours spans contexts. You must configured a queue like below:

```
[6500]
fullname = WaitingQueue
strategy = ringall
timeout = 15
wrapuptime = 15
autofill = no
autopause = no
joinempty = yes
leavewhenempty = no
reportholdtime = no
maxlen = 10
musicclass = default
eventwhencalled = yes
```

Important thing that provide joinempty = yes queue's parameter for leave the new incomming into the queue on waiting.

OdinMonitoring, a call audio recorder RC1 compatible.

Please install samba package for use a windows shared folder, this is a good choise to save disk space" of the iPBX host. Example for debian based OS : apt-get install smbclient cifs-utils

MySql driver must be installed into Windows host. Source ODBC must be created, please check user's rights of the mysql server.

Example to mount a windows shared folder :

```
  mount -t cifs //192.168.3.97/RC1/Db/Data/Audio /home/rc1/ -o username=toto,password=tata,f
```

iPBX dialplan configuration :

Edit the global section into extentions.conf file and ajust or create variables :

```
MIXMON_FASTAGI_ADR=127.0.0.1:4576
RC1_DIR = /home/rc1/
MIXMON_DIR = /var/spool/asterisk/monitor/
```

Follow the sample macro examples.

Macro macro-record-enable can be found into extentions_custom.conf file. This macro initialize the recording process :

```
[macro-record-enable]
exten => s,1,NoOp(macro-record-enable)
exten => s,n,StopMixMonitor()
exten => s,n(check),AGI(agi://${MIXMON_FASTAGI_ADR})
exten => s,n,MacroExit()
exten => s,999(record),Noop(You can playback a voice recording message)
exten => s,1000(record),MixMonitor(${MIXMON_DIR}${MONITOR_CALL_FILE_NAME},b)
;end of [macro-record-enable]
```

Macro macro-record-stop stop recording process and can(must) be called into h extention of the context. If the recording was started the fastagi monitor script inoked and the call recording stopped.

```
[macro-record-stop]
exten => s,1,NoOp(macro-record-stop)
exten => s,n,GotoIf($["${MONITOR_CALL_FILE_NAME}" = ""]?exit:)
exten => s,n,StopMixMonitor()
exten => s,n,AGI(agi://${MIXMON_FASTAGI_ADR})
exten => s,n(exit),MacroExit()
;end of [macro-record-stop]
```

Macro macro-record-rc1-converter can be used for the conversation the asterisk wav file to rc1 compatible format

```
[macro-record-rc1-converter]
exten => s,1,NoOp(record-rc1-converter)
exten => s,n,GotoIf($["${MONITOR_CALL_FILE_NAME}" = ""]?exit:)
exten => s,n,StopMixMonitor()
exten => s,n,System(/usr/bin/sox ${MIXMON_DIR}${MONITOR_CALL_FILE_NAME} -r 8000 -b 8 -c 1 -e
exten => s,n(exit),MacroExit()
;end of [macro-record-enable]
```

Example :

```
 [record-exampe-context]
 include = parkedcalls
 same => n,Macro(record-enable)
 same => n,NoOp(Monitor file :${MIXMON_DIR}${MONITOR_CALL_FILE_NAME})
 same => n,Dial(SIP/6005&SIP/6006,60,tTkK)
```

```
 same => n,Hangup()
 exten = h,1,Macro(record-stop)
 same => n,Macro(record-rc1-converter)
 same => n,NoOp(Monitor file hangup : ${MIXMON_DIR}${MONITOR_CALL_FILE_NAME})
```

Installation

Please transfer odin_[version].tar.gz file into /opt directory of the target host.

Execute tar xvzf odin_[version].tar.gz

Change the directory : cd /opt/odin/install

Execute ./install.sh and follow instructions.

Directory Layout

```
odin/                    --> all application components and library
  client/                  --> web client files

  server/                   --> Nodejs web server

  pyodin/                 --> AMI and F1COM servers

  scripts/                --> Test scripts, not for production

  test/                   --> Test source files and libraries
```

Contact