Raspi Asterisk PBX Alexa Interface

2016-08-05 -- r.grokett

version 1.0

Overview

This project is a proof-of-concept using Asterisk PBX, running on a Raspberry Pi Zero or better, interfaced to Amazon Alexa™ Voice Service API. Using a SIP Phone or SoftPhone, the user dials into their Raspberry Asterisk PBX extension and follows the prompts to speak questions which are sent to Amazon Alexa Voice Service. Responses are spoken back over the phone. No Amazon Echo™ required.

It uses a Perl AGI script and SOX to convert the audio formats and send & receive the AVS JSON API messages over the Internet.   Amazon does not currently charge for this service, but you do have to create an AWS Developer account.

As this is only a proof-of-concept, it does not comply with Amazon's Public/Commercial terms & conditions. But it does meet their developer requirements for private, non-commercial use.

The primary limitation is that you cannot connect an external telephone number to this for use as an inbound service without modification to meet their requirements. (i.e. trigger the recording using a push button (touchtone?). Also, it has not been optimized to reduce overhead and improve response time. But for a single user demo, the Raspi Zero and AVS work quite well.

Requirements

- Raspberry Pi Zero, B+, 2 or 3

- Raspian Jessie installed and up to date
- Raspi ethernet or Wifi connected (direct Ethernet is best)
- SSH access to the Raspberry
- A SIP Phone (Such as Grandstream GXP1620) or SoftPhone such as:
  http://www.asteriskguru.com/tutorials/xlite_softphone.html
  http://www.zoiper.com/en/voip-softphone/download/zoiper-classic

This project assumes a working knowledge of Raspian command-line, SSH and some knowledge of Asterisk PBX. It also assumes you have a Raspberry Pi configured with Raspian Jessie and connected to your network. This does not require a GUI on the Raspberry, so assumes headless operation and a login prompt or SSH access.
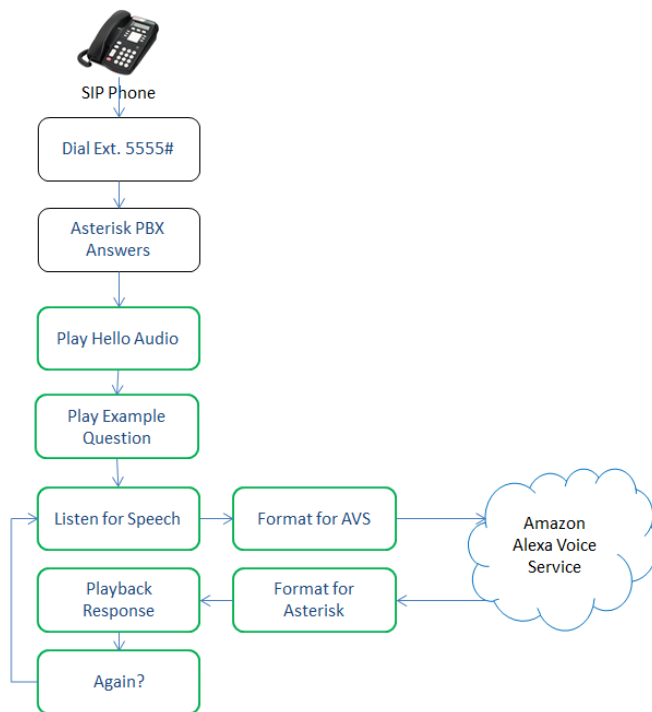
Asterisk PBX Configuration

The Raspberry needs outbound Internet access but the Asterisk PBX does NOT need any external SIP Voice Provider. This project does not use inbound or outbound voice calls.

If you have never used Asterisk, you probably should look up some Asterisk tutorials. Though, most of the information isn't needed here.

This project should be done on a CLEAN, CONFIGURED RASPIAN JESSIE installation. Using an existing installation may have had modifications that impact Asterisk.

1. Install an Asterisk PBX on your Raspberry Pi:
```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install asterisk
```

2. Retrieve the project files from Github:
```
$ cd /home/pi
$ git clone https://github.com/rgrokett/RaspiAsteriskAlexa
```

3. Add a SIP Phone to your Asterisk. Since there are many ways to do this, I am showing a generic configuration. Each phone has different options, but this should be all that is needed here.

   This is adding an extension number 5310 to Asterisk. Remember, this does NOT include inbound or outbound calling. Only internal calls.
   Backup your /etc/asterisk/*.conf files before editing!

   There are snips of these files in the ./RaspiAsteriskAlexa/config/ directory, if you don't want to cut & paste from below.
```
$ cd config
$ sudo cp /etc/asterisk/sip.conf /etc/asterisk/sip.conf.bak
$ sudo cat sip.conf >> /etc/asterisk/sip.conf
$ sudo cp /etc/asterisk/extensions.conf /etc/asterisk/extensions.conf.bak
$ sudo cat extensions.conf >> /etc/asterisk/extensions.conf
```

Append to /etc/asterisk/sip.conf

```
[5310]
type=friend
username=5310
fromuser=5310
host=dynamic
context=local
insecure=port
qualify=500
dtmfmode=rfc2833
disallow=all
allow=ulaw
obtained
progressinband=no
nat=no
mailbox=5310
callerid=5310
```

Append to /etc/asterisk/extensions.conf
```
; Basic SIP Phone
exten => 5310,1,Dial(SIP/5310,15)
exten => 5310,2,Voicemail(5310,u)
exten => 5310,3,Hangup
exten => 5310,102,Voicemail(5310,b)
exten => 5310,103,Hangup
```

```
SIP Phone options:
        SIP Proxy Server: 192.168.1.XX        <- The IP Address of your Raspi Pi  $ hostname -I
        Domain/Realm/Registration: <same IP as above>
        Username: 5310
        Password:  <blank>
```

4. Configure your SIP Phone with the options above. Most other phone options are not used here, so can usually be ignored.

5. Reboot your Raspberry and your SIP Phone and see if you can get dial tone.
   If so, try dialing **1000# on your SIP Phone**. You should get a built-in demo.

   Debugging: If you don't get dial tone, your SIP Phone isn't registering to your Asterisk server.
   Try the following:
   a. Power off and on the Raspberry and then the SIP phone to see if they reconnect.
   b. Verify the files above are edited correctly. Watch for syntax errors!
   c. `$ sudo asterisk -r`
   d. `CLI>  sip show peers`
      This should show your sip phone's IP and status
      Watch out for firewalls, particularly if using a softphone!
   e. Verify IP addresses of the Sip Phone and Raspi and be sure you can ping the phone from the Raspi.
   f. `CLI> sip set debug on`
      This will display SIP connection attempt messages. If you see nothing after a few minutes, then your SIP Phone isn't even trying to talk to Asterisk. You need to dig into the SIP Phone's setup instructions.
   g. Asterisk log messages go to /var/log/asterisk/messages. But note there can be lots of nasty looking messages, most are for unused/unneeded features. So it can be difficult to decipher.

If you have successfully gotten dial tone and the demo, then you now have a working Asterisk PBX!

Alexa/Asterisk Install and Configuration

1. Install required packages:
   You can run the install.sh script to do this.
   `$ cd /home/pi/RaspiAsteriskAlexa/`
   `$ bash ./install.sh`
   This takes a while and will ask questions along the way. Just accept the defaults.

```
# Install packages and files
    sudo cp alexa.agi /usr/share/asterisk/agi-bin/alexa.agi
    sudo cp avs_audio.json /etc/
```

```
        sudo cp sounds/alexa*.sln /usr/share/asterisk/sounds/custom/

        sudo chown asterisk:asterisk /usr/share/asterisk/agi-bin/alexa.agi
        sudo chown asterisk:asterisk /etc/avs_audio.json
        sudo chown asterisk:asterisk /usr/share/asterisk/sounds/custom/alexa*.sln

        cp token.pl /home/pi
        sudo rm /tmp/*
        sudo rm /tmp/token.* /tmp/avs*

        # INSTALL VARIOUS PACKAGES
        sudo apt-get install sox
        sudo apt-get install libsox-fmt-mp3
        sudo apt-get install libwww-perl libjson-perl
        sudo apt-get install flac
        sudo curl -L http://cpanmin.us | perl - --sudo App::cpanminus
        sudo cpanm IO::Socket::SSL --force
        sudo perl -MCPAN -e 'install JSON'
        sudo apt-get install libjson-pp-perl
```

2. Verify the Asterisk /etc/asterisk/extensions.conf contains the below and append if needed:

    ```
    $ sudo tail -30 /etc/asterisk/extensions.conf
    ```

    /etc/asterisk/extensions.conf

    ```
        ; AMAZON ALEXA VOICE
        [alexa_tts]
        exten => 5555,1,Answer()
        ; Get an AWS Token
        exten => 5555,n,System(/home/pi/token.pl)
        ; Play prompts
        exten => 5555,n,Playback(./custom/alexa_hello)
        exten => 5555,n,Playback(./custom/alexa_example)
        ; Alexa API integration
        exten => 5555,n(record),agi(alexa.agi,en-us)
        ; Loop
        exten => 5555,n,Playback(./custom/alexa_another)
        exten => 5555,n,goto(record)
        ; These are not used currently
        exten => 5555,n(goodbye),Playback(vm-goodbye)
        exten => 5555,n,Hangup()
    ```

3. Add the following line to extensions.conf so that the extension is dial-able locally.
    a. Edit /etc/asterisk/extensions.conf
    b. Locate the section called `[local]`
    c. Add a line "**include => alexa_tts**"

    ```
    /etc/asterisk/extensions.conf

    [local]
    ;
    ```

```
; Master context for local, toll-free, and iaxtel calls only
;
ignorepat => 9
include => default
include => trunklocal
include => iaxtel700
include => trunktollfree
include => iaxprovider
include => alexa_tts
```

4. Reboot the Raspberry Pi to pick all these changes.
5. Finished with part 1.

Amazon Alexa Voice Service Setup

You have finished part 1 of the setup. Now you need to create a (free) account on Amazon Alexa Voice Services and get authentication tokens for your Raspi Asterisk.

Excerpted from Getting Started with the Alexa Voice Service:
https://developer.amazon.com/public/solutions/alexa/alexa-voice-service/getting-started-with-the-alexa-voice-service

Register Your Product with AVS

- Sign up for a free Amazon developer account
- Log in to the Amazon Developer Portal
- Select **Get Started** under Alexa Voice Service and follow the instructions to register your product
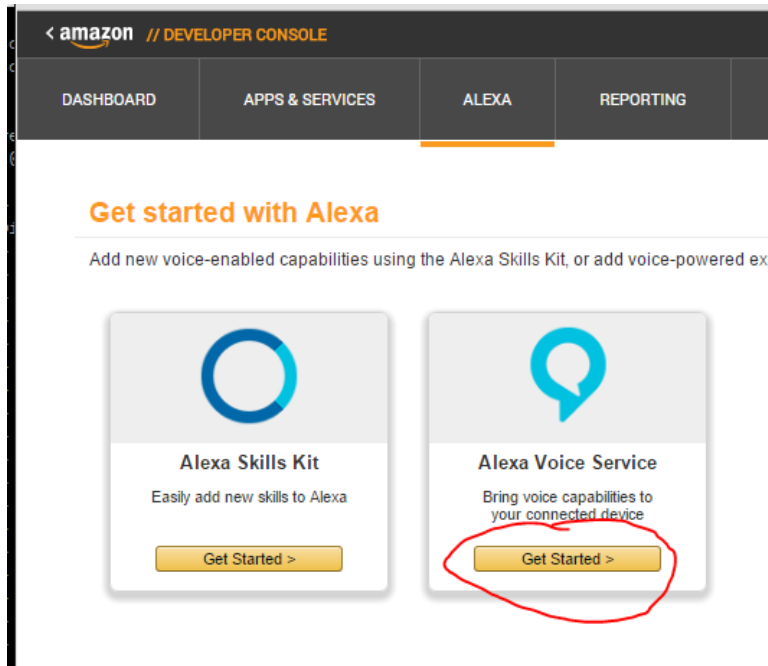
Fig. 1 – Amazon AVS
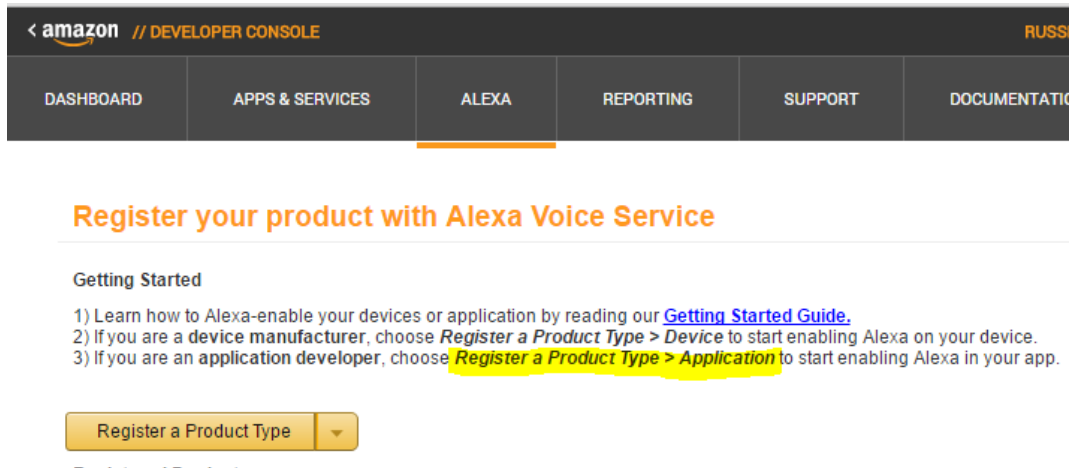
Fig.2 – Register a **Product Type > Application**

Create a new Application Type

* Fields required

Application Type Info ✓	
Security Profile ✓	
Application Details ✓	
Amazon Music ✓	

Company Name
The name of the company you listed in your developer account profile.

Application Type ID *
Choose a unique name that identifies your application. This name will not be shown to end users. It may only contain letters, numbers and underscores.

MyAsteriskPBX

Display Name *
Choose an application name that you want shown to your end users when they register their applications with Amazon.

MyAsteriskPBX

Next

Fig.3 – Fill out the Application Type screen. Use any names you like.
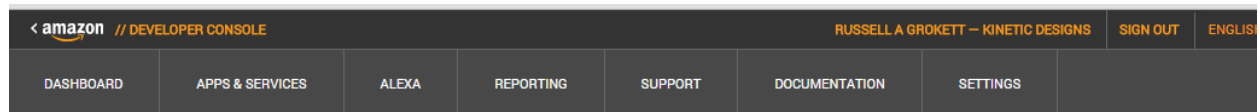
< amazon // DEVELOPER CONSOLE   RUSSELL A GROKETT — KINETIC DESIGNS   SIGN OUT   ENGLISI

DASHBOARD	APPS & SERVICES	ALEXA	REPORTING	SUPPORT	DOCUMENTATION	SETTINGS

Create a new Application Type

* Fields required

Application Type Info ✓	
Security Profile ✓	
Application Details ✓	
Amazon Music ✓	

You need a security profile to identify your application. Your security profile credentials - client ID and client secret - allow your application to securely identify itself to the Alexa Voice Service. If you are building a website, click here to **Learn More**. If you are building an Android or iOS app, click here to **Learn More**.

Security Profile ? *
A security profile is how Amazon identifies your application.

Create a new profile ▼

General	Web Settings	Android/Kindle Settings	iOS Settings

Security Profile Name *
Choose a name for your security profile.

alexa_asterisk

Security Profile Description *
Choose a description for your security profile for Amazon services to use in communicating with you.

Asterisk PBX Alexa

Next

Fig.4 – Create new profile with again, any name desired for the Security Profile Name.

Fig.5 – Security IDs

You will need these IDs for your AGI script. Cut/paste to an empty text file or notepad.

Fig.5b – Add Web Settings

Enter https://localhost

and http://{Your_Raspi_IP_address:5000/code

To find your IP: `$ hostname -I`

The Port number is dummy just for initial setup.

Then click NEXT

Fig.6 – Fill in as desired (this is a non-commercial application)

Enable Security Profile

1. Open a web browser, and visit https://developer.amazon.com/lwa/sp/overview.html

   You may need to log in again and repeat the above URL.

2. Select your security profile:

Fig. 7 – Select your Security Profile

Enter a privacy policy URL beginning with http:// or https://. For this example, you can enter a fake URL such as http://example.com.
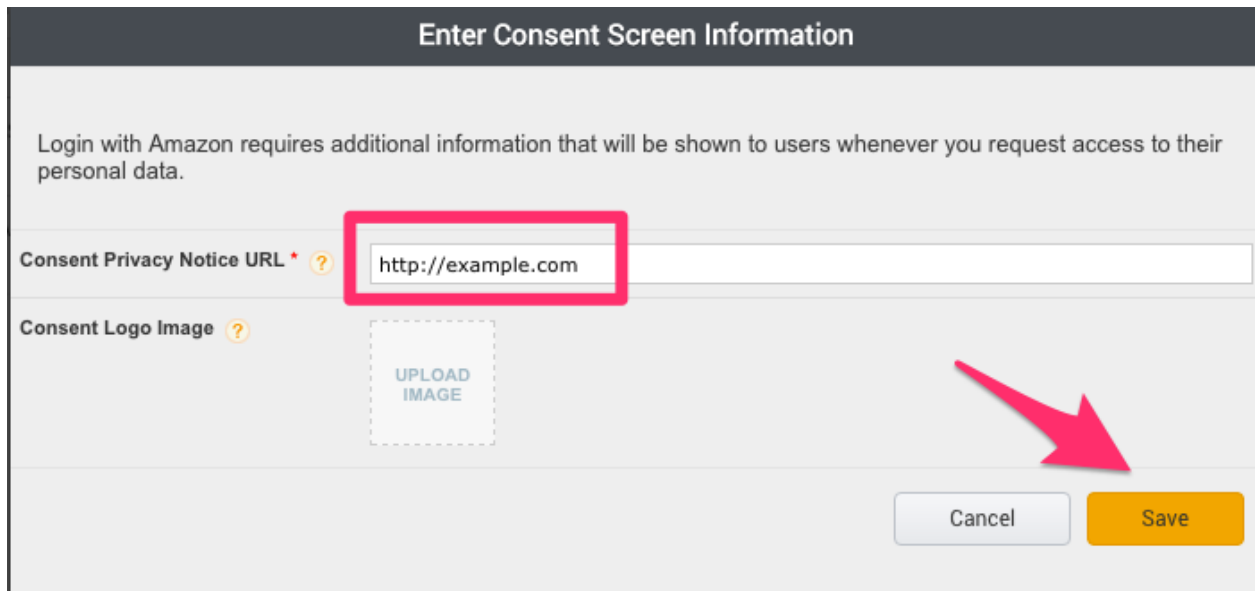

Fig. 8 – Consent Screen (enter a fake url)

3. On the next screen, next to the Alexa Voice Service Sample App Security Profile, click **Show Client ID and Client Secret.** This will display your client ID and client secret. Save these values. You'll need these, if you didn't already get these.

Fig. 9 – Client Credentials

Note: All of the following commands are all on one line each.

Excerpted from: https://developer.amazon.com/public/apis/experience/cloud-drive/content/restful-api-getting-started

4. Copy the URL below to a notepad and EDIT to insert your **client_id** and Paste to your browser.

[https://www.amazon.com/ap/oa?client_id=**YOURCLIENTID**&scope=alexa%3Aall&scope_data=%7B%22alexa%3Aall%22%3A%7B%22productID%22%3A%22kd_raspi_asterisk%22,%22productInstanceAttributes%22%3A%7B%22deviceSerialNumber%22%3A%2212345%22%7D%7D%7D&response_type=code&redirect_uri=https%3A%2F%2Flocalhost](#)
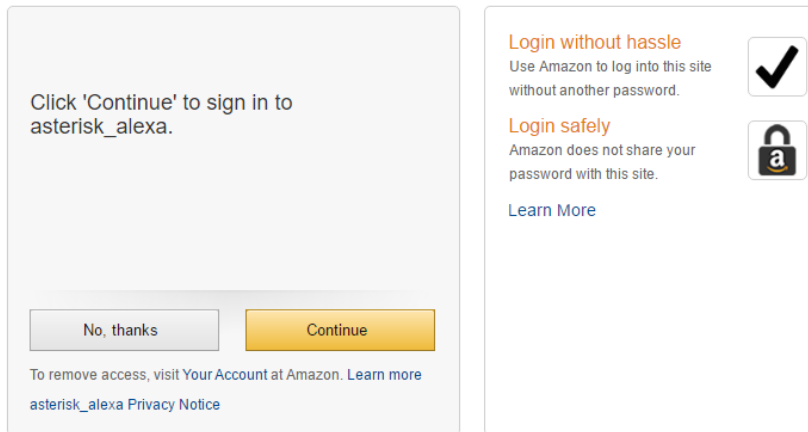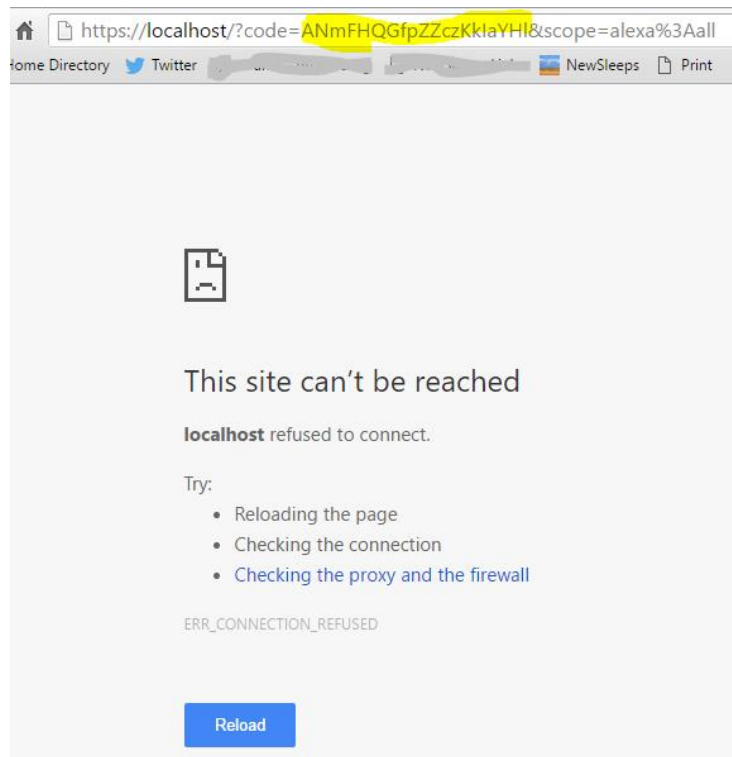
Fig. 10 -- Click Continue

5. You will receive a **browser error message**, but the Address URL should show a **provisioning token**:

    ```
    https://localhost/?code=YOURTOKEN&scope=alexa%3Aall
    ```

Do NOT click anything more. Just copy & paste the URL to your notepad or text file.

6. On your Raspi, edit the file **grant_token.sh** and update the fields and run its cURL command:

```
$ nano grant_token.sh
$ bash ./grant_token.sh

YOURTOKEN      - Example:  ANLdMXNCfOnqxCa…
YOURCLIENTID - Example:  amzn1.application-oa2-client.0b1342a03a674c…
YOURCLIENTSECRET  - Ex:  34c812a9c0601d18f14a7f7b035e6416918d…
```

You should receive back a big JSON message with an ACCESS TOKEN and a REFRESH_TOKEN.

7. Your access_token will be used by Asterisk every time it makes a call to Amazon API. But this token expires every hour, so Asterisk will be calling a program called **token.pl** each time you dial the Alexa extension number.

Your refresh_token never expires, but is only good for retrieving a new Access Token. It is used by token.pl to retrieve a new Access Token.

NOTE: *Pay close attention to copying and pasting these two tokens. They are each one long line (no newlines).*

8. Cut & Paste them into your notepad or text file and save the file for future reference.

```
Example:
{      "access_token":"Atza|IQEBLzAtAhRqd3LSY6n_A_VERY_LONG_STRING…",
"refresh_token":"Atzr|IQEBLjAsAhQgCKZ5Ind88BUAgdO9k7_ANOTHER_VERY_LONG_STRING",
       "token_type":"bearer","expires_in":3600
}
```

9. Edit the **token.pl** perl program inserting your Refresh Token and Client Secret. Then copy the token.pl to /home/pi/token.pl

```
$ nano token.pl
$ cp token.pl /home/pi/
```
Be sure to cut/paste the RIGHT TOKENS! And don't cut off or alter them.

```
# Send POST Request
my $post = "grant_type=refresh_token&refresh_token=
YOUR_VERY_LONG_REFRESH_TOKEN&client_id=YOURCLIENTID&client_secret=YOURCLIENTSECRET" ;
```

Good News; Since the Refresh Token and Client Secret never expire, you only have to do all of these steps once.

NOTE: If you get errors, you can start again at Step 4, above (See. Fig.9 Client Credentials) and repeat the process. Look at the Amazon document https://developer.amazon.com/public/apis/experience/cloud-drive/content/restful-api-getting-started

10. Run the token.pl This will query Amazon OAuth2 and return a JSON which is parsed into the /tmp directory.

```
$ perl /home/pi/token.pl
```

Look in /tmp for the response files:
```
$ ls –l /tmp/token.*

-rw-rw---- 1 pi pi    439 Aug  4 13:39 token.avs
-rw-rw---- 1 pi pi    685 Aug  4 13:39 token.req
-rw-rw---- 1 pi pi    932 Aug  4 13:39 token.resp
```

You should view the files. You should NOT see any error message in them. The files will contain a new Access Token as well as the cURL request and response messages.

```
$ sudo cat /tmp/token.resp
```

Example of a good token.resp

```
{"access_token":"Atza|IwEBILhKrSN8kfzozbImVfr6AAySL_kzVfEvamrA772hjH_Zvx-
0IIIgjXNRK4tXvT6tkaLJ6kSh_F9UbDK3bU-iEfXxOA68jhGEhC3Vaw96pDrOvcuv29rL5Hxhte5-
zWkzf-sL5il5PtuezNKWuPCvjpFCdB5Tm1a6HaiebPk9cDWosHZkFLYVhvK5…
","refresh_token":"Atzr|IwEBIM617be3fOqudYsrfi9KXimW6432DWIgCptd-
gGqFvnUOuXgN4cJ4l8uvzaQM4Ozoh-X-
Nf1wgcgprrG4cr0P2mmfrNBgICaUHtc0lt8Ra4Y31QsuElMhIrQWbzz3e0hWcI3xGprwhBEB6Yx6J
43sAZnFUwrBo0QZ_iDJSWd7c3JqK5LtbzgDeczYn2M3pcbAyNJG8r3bj8zG0Q2v9k_BwGKXCDAe6T
ITse0YP_N89KCrmGP4WZPFGGrqAuOA8DAJSoKMGneqDLIX0wBc5R08xLNNnyYbGOQ…
","token_type":"bearer","expires_in":3600}
```

If you see an error, turn on debug mode (below) in the **token.pl** and rerun the command. Be sure to VERIFY that you have cut&pasted the FULL refresh token, client ID and client Secret. Obviously, if these are wrong, Amazon API will return Authentication errors. Watch out for improper carriage returns in the tokens (there are none!) due to word wraps or cut/paste errors. Also use the correct tokens in the proper places. 99% of the problems will be related to the tokens.

```
        # Send POST Request
        my $debug = 1;
```

11. Delete the /tmp/token.* files after you have it working so that Asterisk can write new files.

```
$ sudo rm /tmp/token.*
```

You are now ready to try out your Asterisk Alexa interface.

12. Reboot your Raspberry Pi.

13. Using your SIP Phone, dial 5555#
    You should dial tone, ringing and then audio prompts.
    If you do, GREAT! Try any questions to Alexa:
    - What is the weather in Atlanta, Georgia?
    - How big is the Earth?
    - What time is it in London, England?
    - Tell me a Cat Joke
    - Etc…

Just hang up when done. It will reset automatically.

TROUBLESHOOTING

1) You did get dial tone and able to call the Asterisk demo (1000#) didn't you?
2) Access Asterisk CLI: $ sudo asterisk –r
3) CLI> agi set debug on
4) CLI> core set debug 4
5) Make a call to 5555# again and watch the messages.
6) You can look at **/home/pi/RaspiAsteriskAlexa/alexa.agi** for matching debug messages to code.
   Note that Asterisk displays lots of extra debug info beyond the messages

   If you make changes to the Asterisk extensions.conf or sip.conf, you should restart Asterisk
   CLI> core restart now
   If you make changes to alexa.agi, you must copy it over to the AGI directory:
   ```
   $ sudo cp alexa.agi /usr/share/asterisk/agi-bin/alexa.agi
   ```

7) Most likely issue is the Amazon Authentication failing. If the tokens are wrong, then it will not work! Go back to the token section to verify.
8) If the phone rings w/o answer, then extensions.conf has a problem in the 5555 section. You did add the `include => alexa_tts` statement?
   If you edit extensions.conf, you must reload using CLI> dialplan reload
9) If the phone answers but no audio, verify that the files in the asterisk directories are owned by asterisk userid:
   ```
   sudo chown asterisk:asterisk /usr/share/asterisk/agi-bin/alexa.agi
   ```

```
sudo chown asterisk:asterisk /etc/avs_audio.json
sudo chown asterisk:asterisk /usr/share/asterisk/sounds/custom/alexa*.sln
```

10) If you get no response back from Alexa for your question, check the Amazon authentication again. Be sure you can ping `access-alexa-na.amazon.com` from your Raspi.
11) If all else fails, step back thru the commands above to be sure you didn't miss a step! There are many steps, so it's easy to accidentally skip one.

Results

I have found that the speech recognition of Amazon Alexa Voice Service is excellent when used with a SIP telephone and Asterisk PBX. The recognition appears to be quite good whether on a handset or speakerphone. The Raspberry Pi Zero is able to handle the Asterisk audio processing quite well, even when using WiFi. Of course, the wide range of question/answers that Amazon Alexa can handle is equivalent to Amazon Echo™ , without the location and remote control services.

Next steps would be to pursue whether Amazon Voice Service API supports Amazon Lambda or IFTTT services for skills and remote control functionality.