

Configuración HA y Alexa para plugin haaska

Requisitos

- Home assistant (HA) con versión superior a la 0.62
- Acceso a HA desde fuera de LAN y mediante https (uso duckdns, ddns, ssl, proxy-reverse)
- Máquina linux (VM, raspi o docker) instalado zip y python-pip
- Cuenta amazon desarrollador (en el manual se explica como crearla)
- Archivo de texto preparado para apuntar todos los tokens, id secrets, etc.

Referencias:

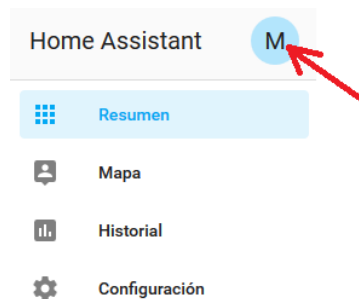
- <http://collingwood.me.uk/blog/index.php/hass-io-and-alexa/>
- <https://github.com/mike-grant/haaska/wiki/Setup>

Procedimiento:

1.- En el *configuration.yaml* de HA se tiene que tener esto puesto y reiniciamos HA

```
api:  
  
alexa:  
  
smart_home:
```

2.- Se ha de generar un token en HA, para ello en el menú de la izquierda donde aparece la inicial, clicar y vamos a la última opción "Tokens de acceso de larga duración" y creamos el token (lo apuntamos en el archivo de texto)



Tokens de acceso de larga duración

Crea tokens de acceso de larga duración para permitir que tus scripts interactúen con tu Home Assistant. Cada token será válido por 10 años desde la creación. Los siguientes tokens de acceso de larga duración están actualmente activos. [Aprenda a realizar solicitudes autenticadas.](#)

[CREAR TOKEN](#)

3.- Nos descargamos el archivo zip (<https://github.com/mike-grant/haaska/archive/master.zip>) directamente o mediante git, para ello puede ser necesario instalarlo:

```
$> apt-get install git
```

posteriormente

```
$> git clone https://github.com/mike-grant/haaska.git
```

4.- En la máquina linux con zip y python-pip instalado (si no lo tienes debes de ejecutar: apt-get install zip python-pip). Descomprimos el zip y dentro de la carpeta config, hay un fichero llamado *config.json.sample*. Hay que renombrar ese fichero a *config.json* y editar su contenido con los siguientes datos:

```
{  
  "url": "https://url_acceso_ha_wan:puerto/api",  
  "bearer_token": "aqui dentro el token de HA del punto 2",  
  "debug": true,  
  "ssl_verify": true,  
  "ssl_client": []  
}
```

5.- Una vez editado, hacemos:

```
$haaska/config/> cd ..
```

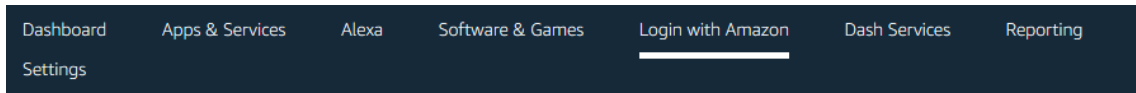
```
$haaska/> make
```

Se nos habrá generado un archivo zip llamado *haaska.zip* y lo guardamos en nuestro PC para posteriormente subirlo a amazon.

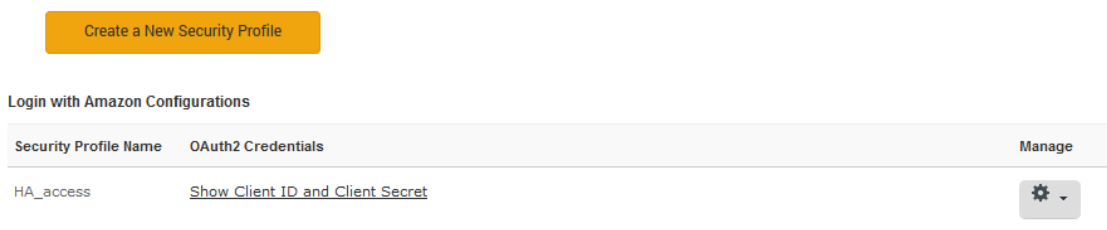
6.- Vamos a la siguiente url, si no tienes cuenta, create una

<https://developer.amazon.com/home.html>

En el menú de arriba, vamos a la opción "Login with Amazon"



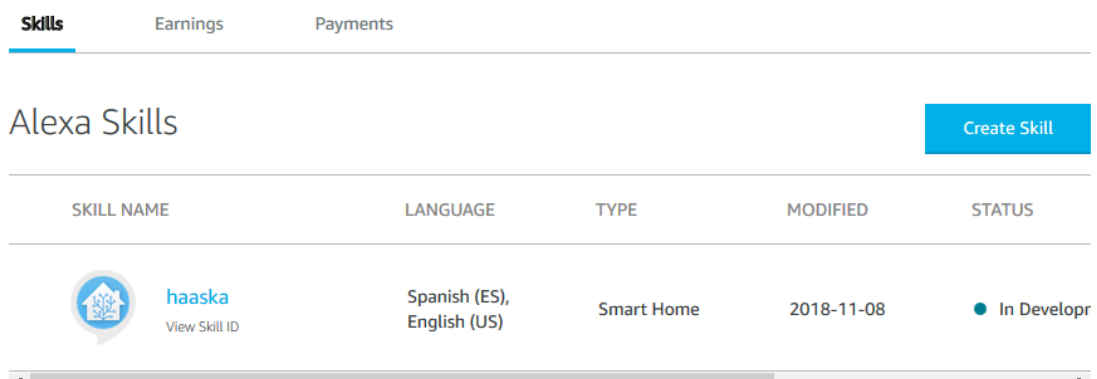
Y creamos un nuevo perfil de seguridad "Create a New Security Profile", poniendo nombre y dándole a guardar.



Ahí se nos habrán creado 2 token , Client ID y Client Secret, los anotamos.

7.- Ahora necesitamos crear nuestro skill, para ello en el menú del punto 6, vamos a Alexa --> Alexa Skills Kit

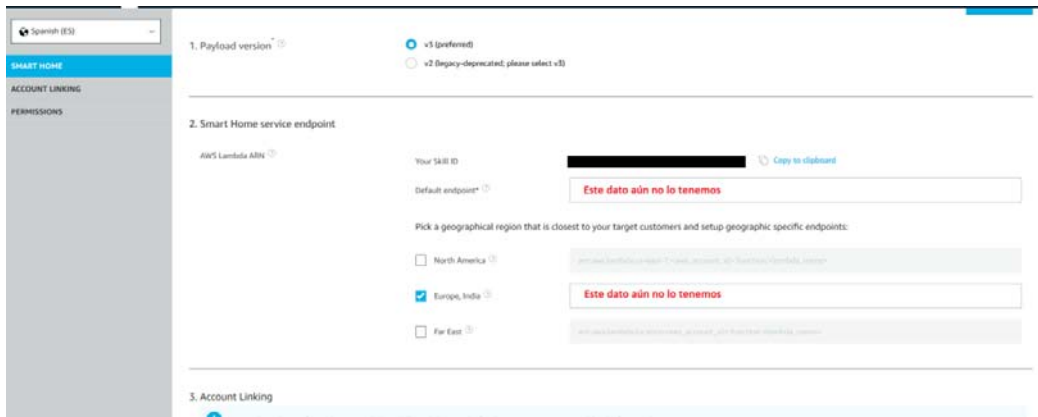
<https://developer.amazon.com/edw/home.html#/skills>



7.- Le damos a "Add New Skill Button" (Añadir nuevo Skill)

8.- En la opción de "SkillName" le ponemos el nombre haaska. En el menú de la izquierda, seleccionamos SMART HOME guardamos y copiamos el código Application ID en nuestro archivo.

9.- La ventana de abajo la completamos como en la imagen de abajo (ATENCIÓN CON EL LENGUAJE QUE PONEMOS QUE MÁS TARDE NOS AFECTARÁ)



Ahora debemos crear nuestra función LAMBDA y que nos generará la información que falta más arriba y a su vez el código python para interactuar con nuestro HA.

10.- Vamos a esta url para crear nuestra función Lambda, si no tienes cuenta debes de crearla. Además tienes que dar detalles de una cuenta, pero no te hacen ningún cargo.

<https://console.aws.amazon.com/lambda/home?region=us-east-1>

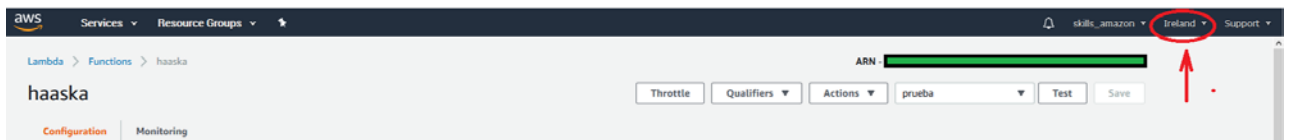
Si antes no has usado la Lambda tienes que crear un IAM lambda Basic Execution, para ello, sigue este manual de Amazon (A partir del apartado Add a Lambda Function).

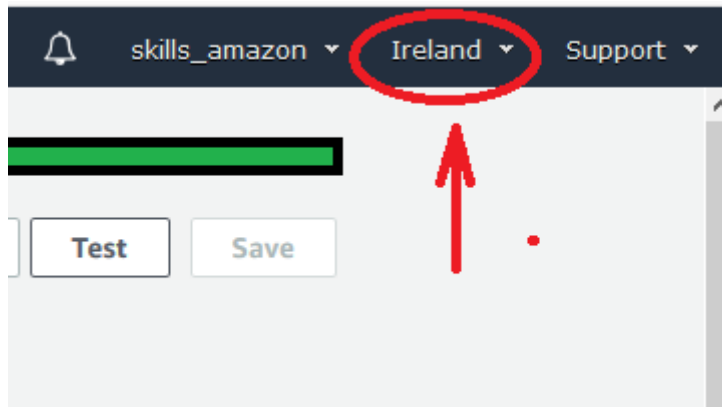
<https://developer.amazon.com/docs/smarthome/steps-to-build-a-smart-home-skill.html>

También puedes seguir el primer manual de las referencias de este documento a partir del apartado 6.6 y ES IMPORTANTE SELECCIONAR EL IDIOMA DE LAMBDA FUNCTION ACORDE AL LENGUAJE DE TU SKILL según la tabla de abajo, para ello arriba a la derecha tenemos que seleccionar esta opción de lenguaje.

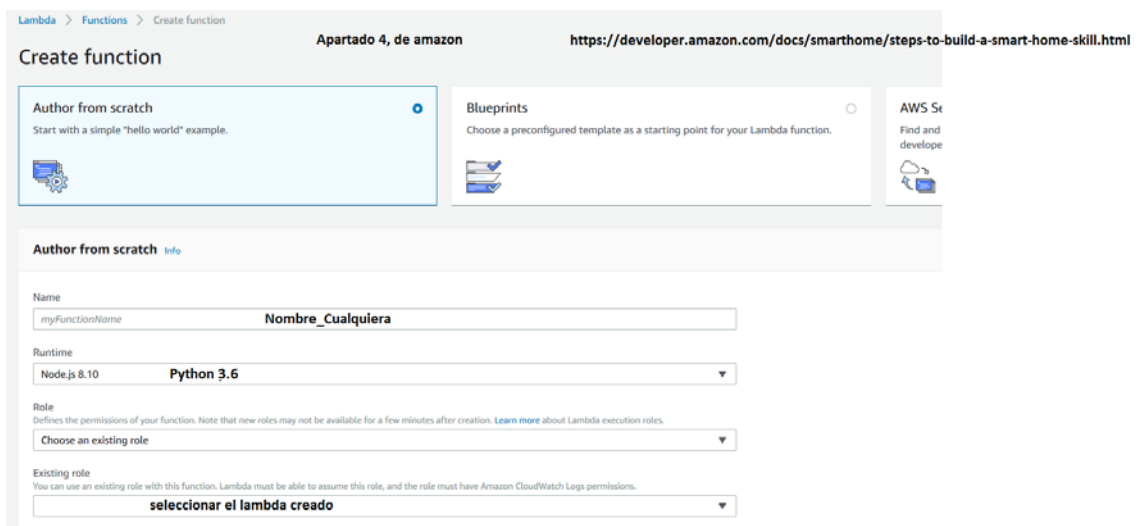
Skill language	Endpoint Region	Lambda Function Region
English (US), English (CA)	North America	US East (N. Virginia)
English (UK), French (FR), German, Italian, Spanish (ES)	Europe, India	EU (Ireland)
English (IN)	Europe, India	EU (Ireland)
Japanese, English (AU)	Far East	US West (Oregon)

Arriba a la derecha de la ventana lambda. En mi caso:

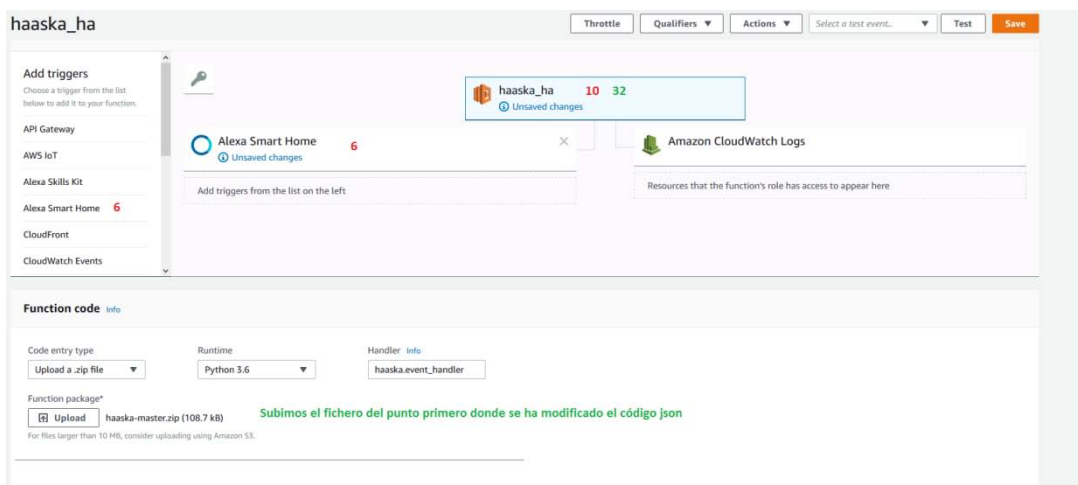


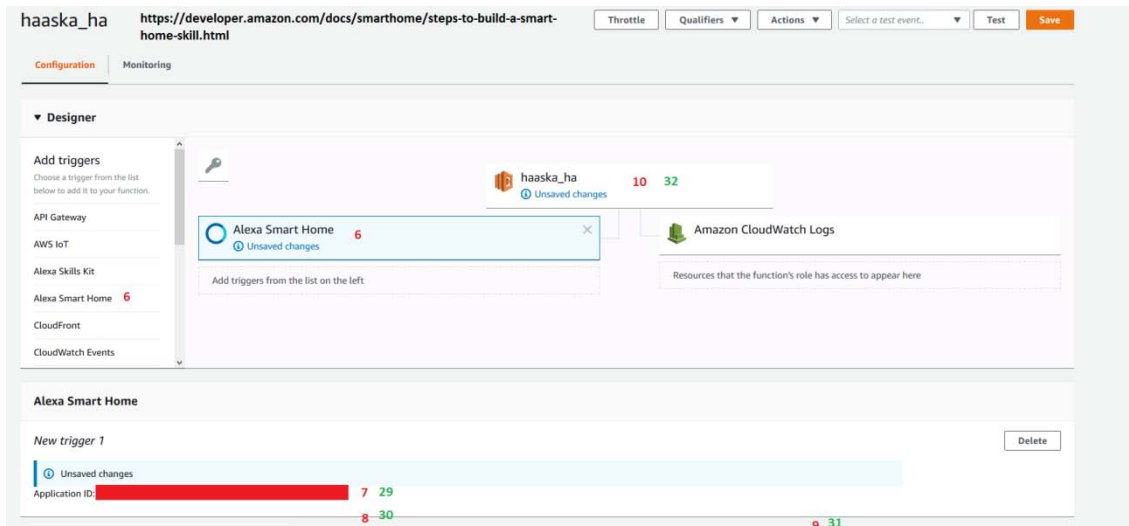


Cuando estemos ya en Lambda, tenemos que hacer esto:



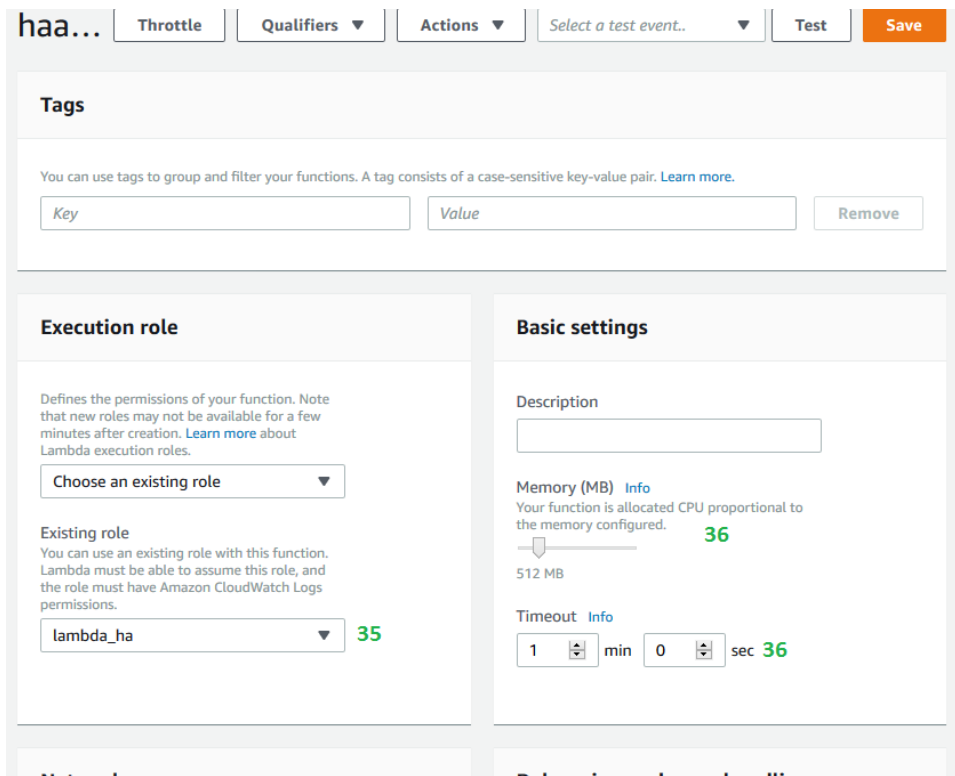
11.- Aquí debemos de subir nuestro fichero haaska.zip y en el Handler, escribimos haaska.event_handler



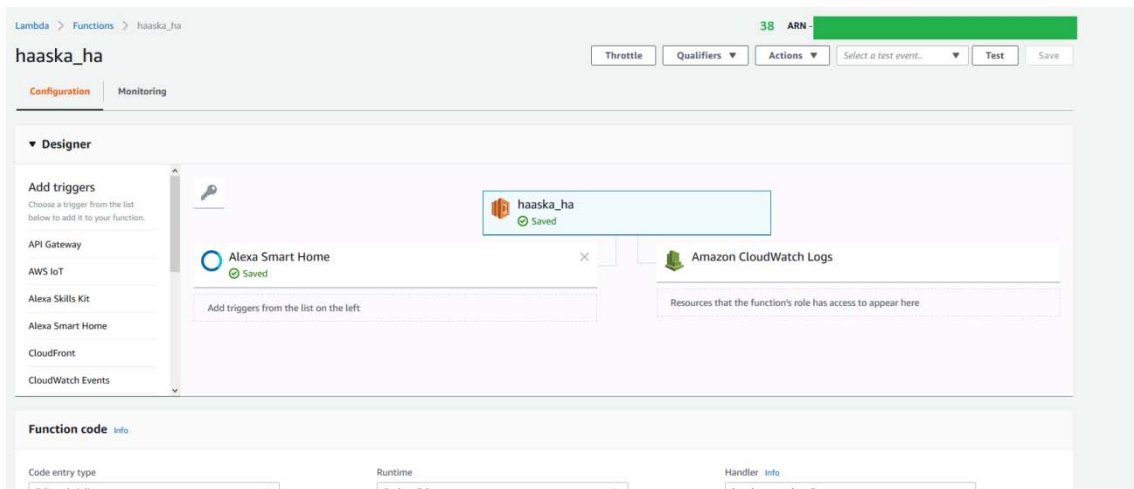


For "Role":

- Select "Choose an existing role", and underneath, select `lambda_basic_execution` if it exists
- If `lambda_basic_execution` doesn't exist, select "Create a custom role" instead, and enter `lambda_basic_execution` as the "Role Name"



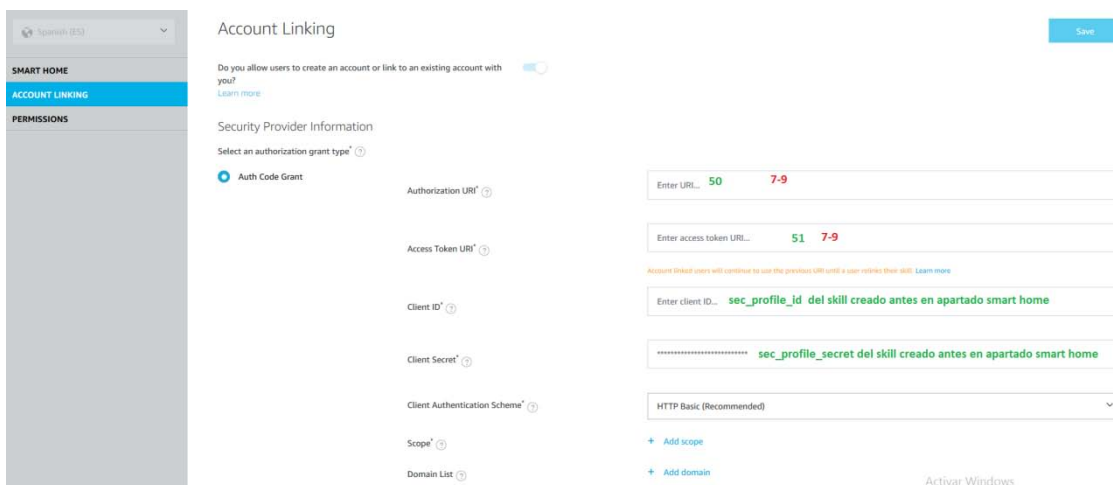
12.- Una vez hayamos dado a "save" tendremos nuestro código ARN en la parte superior derecha y que es el necesario para rellenar el punto 9 del documento



13.- Volvemos a la web de nuestro Skill y en el apartado "Account Linking" añadimos estos datos:

Under the "Account Linking" section:

- Set Authorization URL to: <https://www.amazon.com/ap/oa>
- Set the Client ID to the previously noted value from Login with Amazon
- Set Scope to: `profile`
- Set Access Token URI to: <https://api.amazon.com/auth/o2/token>
- Set Client Secret to the previously noted value from Login with Amazon
- Note the one or more "Redirect URL(s)"



Select an authorization grant type [?]

Auth Code Grant

Authorization URI [?]

Access Token URI [?]

Account linked users will continue to use the previous URI until a user re-links their skill. [Learn more](#)

Client ID [?]

Client Secret [?]

Client Authentication Scheme [?]

Scope [?] X

[+ Add scope](#)

[+ Add domain](#)

Domain List [?]

Default Access Token Expiration Time [?]

Redirect URLs [?]

[Anotamos las urls](#)

Activar Windows

14.- En la web de skills, vamos a Distribution -->Skill Preview y ponemos:

- Public Name: haaska
- Enter some Example Phrases
- Category: Smart Home
- Go to Launch > Privacy & Compliance and select No for all questions and tick Export Compliance.

alex developer console

< Your Skills haaska Build Test Distribution Certification Analytics


Skill Preview [?]

English (US)

Spanish

Privacy & Compliance

Availability

 **Spanish Store Preview**

Tell us how your skill should appear to Spanish customers.

Public Name [?]

6/50 characters

15.- Ahora vamos a editar el Perfil de seguridad que hemos creado en el punto 6. Así que vamos a Login with Amazon y en nuestro perfil le damos al icono de la derecha. En Web Settings añadimos las urls del punto 13.

HA_access - Security Profile

General **Web Settings** Android/Kindle Settings iOS Settings TVs and Other Devices Settings

To use Login with Amazon with a website, you must specify either an allowed JavaScript origin (for the Implicit grant) or an allowed return URL (for the Authorization Code grant). [Learn More](#)

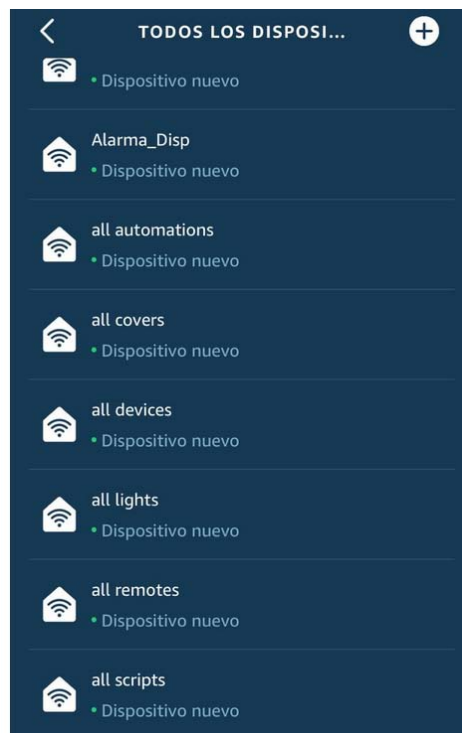
Client ID	[REDACTED]
Client Secret	[REDACTED]
Allowed Origins ?	
Allowed Return URLs ?	[REDACTED]

Añadimos las urls que se generaron en nuestro skill

Edit

16.- Po último vamos a nuestra app de alexa y en Skill activamos nuestra nueva skill llamada haaska, buscamos dispositivos y ver el resultado.

Si te ha funcionado, en dispositivos veras todos los que tienes dados de alta en HA



Activar Dormir Semana

[Editar el nombre](#)

ACERCA DE

Conectado a través de
Home Assistant

Descripción
automation.activar_dormir_semana

Tipo
Otro

Activado

