

## Alexa Payload V3 Plugin für SHNG:

Da die Installation des Plugins mit all seinen zusätzlichen Abhängigkeiten doch etwas mehr ist, hier mal eine Anleitung mit den folgenden Voraussetzungen:

- SHNG Image 1.6 für den RaspberryPi von Onkelandy
- Router ist eine Fritzbox
- DynDNS Account (z.B. no-ip). Ein MyFritz Zugang wird nicht funktionieren! Der DynDNS Zugang muss vorhanden sein, oder erst erstellt werden bevor wir anfangen. Wir benötigen hier später die Hostadresse.
- Amazon Account ist vorhanden.

Das Plugin benötigt, im groben, 8 Dinge die gemacht werden müssen:

1. In der Fritzbox müssen Ports freigegeben werden (443 und 80).
2. In der Fritzbox DynDNS Freigeben (z.B. no-ip.com).
3. In SHNG muss das Plugin aktiviert werden und die Items konfiguriert werden.
4. NGINX auf dem RaspberryPi so einstellen, dass ein Reverse Proxy aktiv ist und ein öffentliches Zertifikat vorhanden ist.
5. Es muss ein Amazon developer Account, mit dem Account-Namen der auch die Alexa-Geräte betreibt, erstellt werden.  
( <https://developer.amazon.com/home.html> )
6. Ein neuer Alexa Skill wird erzeugt  
( <https://developer.amazon.com/alexa/console/ask> )
7. Ein Amazon AWS Datenbank Account muss erstellt werden  
( <https://aws.amazon.com> ) - Lambda-Service – Hier kann ein beliebiger Account gewählt werden (Muss nicht der gleiche sein wie beim developer Account).
8. In der Alexa APP wird der neu erstellte Skill aktiviert und verbunden.

Fangen wir mit den einfachen Sachen an:

### 1: Fritzbox Ports freigeben:

**Achtung!** Wenn ein Fritzbox Fernzugang auf der Fritzbox aktiviert ist, muss dieser erst einmal deaktiviert werden. Ansonsten kann kein Port 443 freigegeben werden, da der Fernzugang diesen Port für sich beansprucht. Also als erstes gucken unter: **Internet / Freigaben / Fritzbox-Dienste** ob dort bei „**Internetzugriff auf die FRITZ!Box über HTTPS aktiviert**“ ein Haken ist. Wenn dort ein Haken ist, diesen entfernen und mit „übernehmen“ übernehmen.

- Auf der Fritzbox Oberfläche den Reiter **Internet / Freigaben / Portfreigaben** wählen.
- **Gerät für Freigabe hinzufügen** drücken.
- In der Spalte **Gerät**, den SHNG RaspberryPi wählen.
- Dann ganz unten auf „**Neue Freigabe**“ klicken und Das neue Fenster so wie im Bild ausfüllen

**Freigabe anlegen**

MyFRITZ!-Freigabe  
 Portfreigabe

Anwendung:

Protokoll:

Port an Gerät:  bis Port

Port extern gewünscht (IPv4):

---

Freigabe aktivieren

- Mit **OK** bestätigen
- Dann das gleiche mit Port 80 durchführen („**Neue Freigabe**“, **Portfreigabe**, **Anwendung: „http-Server“**, **OK**)

Freigaben

Status	Bezeichnung	Protokoll	IP-Adresse im Internet	Port extern vergeben	
●	HTTPS-Server	TCP	██████████	443	<input type="button" value="✎"/> <input type="button" value="✕"/>
●	HTTP-Server	TCP	IPv4	80 (80)	<input type="button" value="✎"/> <input type="button" value="✕"/>

---

- Das Ganze dann aktivieren mit „OK“

## 2: Fritzbox DynDNS Anlegen:

Dann muss noch der DynDNS Zugang in der Fritzbox angegeben werden.

Unter **Internet / Freigaben / DynDNS** die Zugangsdaten zu dem eigenen DynDNS Account, wie im Bild, eintragen. Hier ist der Domainname der festgelegte Hostname des no-ip Accounts.

Internet > Freigaben ?

Über DynDNS können Anwendungen und Dienste, für die in der FRITZ!Box-Firewall Portfreigaben eingerichtet wurden, unter einem festen Domainnamen aus dem Internet erreicht werden, obwohl sich die öffentliche IP-Adresse der FRITZ!Box mit jeder Interneteinwahl ändert.

DynDNS benutzen

Geben Sie die Anmeldedaten für Ihren DynDNS-Anbieter an.

DynDNS-Anbieter:

Domainname:

Benutzername:

Kennwort:

Damit sind die Einstellungen in der Fritzbox abgeschlossen!

### 3: In SHNG das Alexa Plugin aktivieren und die Items einrichten:

Um das Plugin in SHNG zu aktivieren, einfach den folgenden Code in der plugin.yaml hinzufügen:

```
Alexa4P3:  
  class_name: Alexa4P3  
  class_path: plugins.alexa4p3  
  service_port: 9000
```

Um ein Beispiel Item einzurichten, bitte in der eigenen item.yaml den folgenden Code bei einem schaltenden Item hinzufügen (z.B. ein Lampe im WZ)

```
alexa_name: Wohnzimmer Licht  
alexa_description: Licht im Wohnzimmer  
alexa_device: Licht_wz  
alexa_actions: TurnOn TurnOff  
alexa_icon: LIGHT
```

Also zum Beispiel so:

```
Wohnzimmer_Licht:  
  
  type: bool  
  visu_acl: rw  
  knx_dpt: 1  
  knx_send: 2/7/12  
  knx_init: 2/7/13  
  alexa_name: Wohnzimmer Licht  
  alexa_description: Licht im Wohnzimmer  
  alexa_device: Licht_wz  
  alexa_actions: TurnOn TurnOff  
  alexa_icon: LIGHT
```

### 4: NGINX auf dem RaspberryPi so einstellen, dass ein Reverse Proxy aktiv ist und ein öffentliches Zertifikat vorhanden ist:

*Erst einmal vorab: Da ich das auf meinem RPI nicht mehr nachstellen kann, da ich ja schon alles erstellt habe und jetzt gefragt werde ob ich die alten Einstellungen ersetzen möchte, sieht das bei mir nun anders aus als bei der Ersteinrichtung. Deshalb hier sozusagen ein Gedächtnisprotokoll. Sollte aber so oder so ähnlich passen;-)*

- Bitte mit einem Terminal Programm (z.B. Putty) per ssh mit dem RPI verbinden.
- Auf der Kommandozeile „**setup\_all**“ eingeben.
- Solange die Möglichkeiten mit „**Skip**“ überspringen, bis die Frage nach einem NGINX Reverse Proxy kommt.

```
You can setup nginx as a Reverse Proxy to securely access the  
To work correctly you need to forward port 443 in your router  
Furthermore you need to activate a Dynamic DNS service on your  
1) Enable  
2) Disable  
3) Skip  
#?
```

- Hier mit 1 bestätigen.
- Anschließend die Schlüssel erstellen (Den Ländercode z.B. „DE“ Die Stadt in der der eigene Server steht z.B. „Braunschweig“, die eigene Email Adresse z.B. [Mustermann@gmx.de](mailto:Mustermann@gmx.de) und die Hostadresse/Domain aus dem erstellten DynDNS Account:

```

1) Create new keys
2) Show directory content
3) Skip
#? 1
If you had a previous pki folder it got copied to pki_backup.

You have setup the variables for key generation like this:
set_var EASYRSA_REQ_COUNTRY    "DE"
set_var EASYRSA_REQ_CITY      "W"      "]"
set_var EASYRSA_REQ_EMAIL     "ε"      " :@gmx.de"
set_var EASYRSA_REQ_CN        "ε"      ".hopto.org"
set_var EASYRSA_BATCH         "yes"

```

- Dann weiter dem Dialog folgen und Serverzertifikate erstellen lassen.
- Bei der Frage nach einem **Private Key** Password bitte ein Passwort eingeben und im zweiten Schritt noch einmal bestätigen.
- Bei weiteren Fragen nach einem Passwortschutz kann ruhig **2** also kein Passwort gewählt werden.
- Dann wird noch gefragt ob man ein **Client Zertifikat** erstellen möchte (z.B. für den eigenen PC um mit dem Zertifikat auf den Reverse Proxy zugreifen zu können). Hier kann man irgendeinen Namen angeben unter dem das Zertifikat dann erstellt wird.

```
Please define the name of your client (string like MacBook, iPhone, etc.). Hit Enter to create no (more) client certificates. █
```

- Es wird nun gefragt, ob noch weitere Client Zertifikate erstellt werden sollen. Hier einfach mit einem einfachen „**ENTER**“ beenden.
- Anschließend muss noch das **Private Key Password** in ein lua Script eingetragen werden. Das wird in diesem Schritt automatisch erledigt wenn man das vorher festgelegt Passwort hier angibt und Enter drückt.

```
You have to put your private key password in the lua script to make reverse proxy work correctly.
Either do it manually by changing first line in /etc/nginx/scripts/hass_access.lua.
Or provide the password here and let me insert it automatically (Hit enter to skip): █
```

- Nun wird noch abgefragt ob ein **letsencrypt Zertifikat** erstellt werden soll. Das wollen wir, das ist das was wir letztendlich für das Alexa Plugin benötigen.

```
Creating Letsencrypt certificate
```

- Es wird noch gefragt, ob man die Verschlüsselung durch die Stärkeren Diffie-Hellman-Parameter verstärkt werden soll. Wer das nicht möchte oder hier Probleme beim Erstellen hat (Dauert sehr lange!), kann diese Option weglassen, muss dann aber in der Datei **https.conf** im Ordner **/etc/nginx/conf.d/** diese Zeile auskommentieren: **ssl\_dhparam /etc/ssl/certs/dhparam.pem;**

Also mit: **sudo nano /etc/nginx/conf.d/https.config** die Datei bearbeiten und die Zeile so ändern: **# ssl\_dhparam /etc/ssl/certs/dhparam.pem;**

- Wenn das alles ohne Fehler durchgelaufen ist, haben wir auf der RaspberryPi Seite alles erledigt was wir für das Alexa Plugin benötigen.

**4a:** Halt, wir müssen nun noch einen User:Password für Alexa im NGINX festlegen.

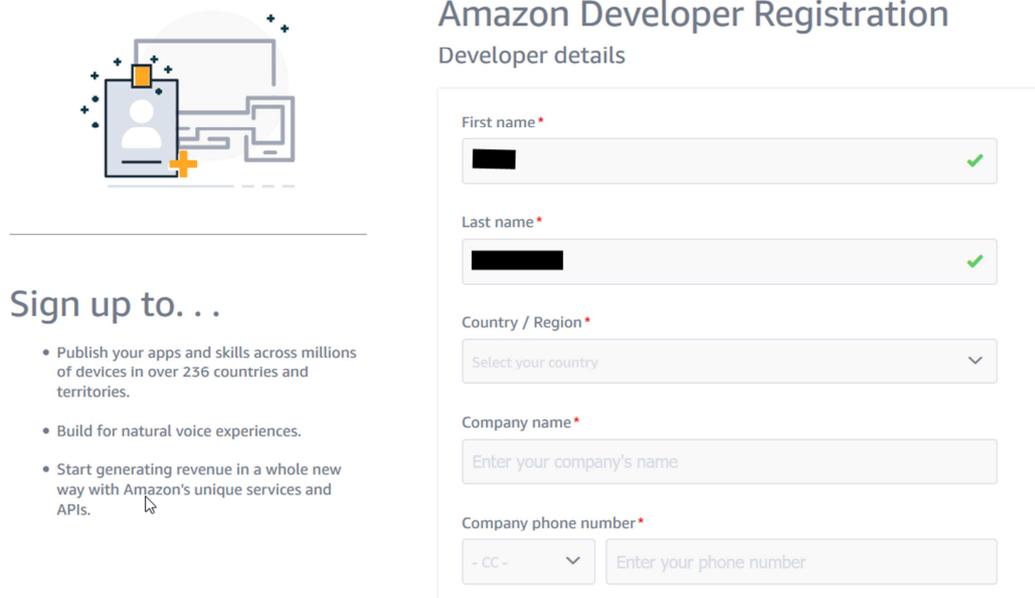
- Das machen wir in der Datei „**/etc/nginx/.alex**a“ mit folgendem Kommando auf der Kommandozeile (<Username> gegen den eigenen ersetzen)

```
sudo htpasswd -c /etc/nginx/.alex <username>
```

- Den Usernamen und das Passwort bitte auch irgendwo aufschreiben, das benötigen wir auch noch einmal später in der Lambda Funktion!

## 5: Eine Amazon developer Account erstellen:

- Bei der Adresse: <https://developer.amazon.com/home.html> mit dem eigenen Amazon Benutzernamen anmelden und auf dieser Seite die Daten mit dem roten Stern vervollständigen und mit „**Submit**“ weiter gehen.



The image shows a screenshot of the Amazon Developer Registration page. On the left, there is a graphic with a person icon and a plus sign, and the text "Sign up to . . .". Below this, there are three bullet points: "Publish your apps and skills across millions of devices in over 236 countries and territories.", "Build for natural voice experiences.", and "Start generating revenue in a whole new way with Amazon's unique services and APIs." On the right, the "Amazon Developer Registration" form is visible, titled "Developer details". It contains several input fields: "First name" (with a green checkmark), "Last name" (with a green checkmark), "Country / Region" (a dropdown menu with "Select your country" and a downward arrow), "Company name" (with the placeholder "Enter your company's name"), and "Company phone number" (with a dropdown menu for country code and the placeholder "Enter your phone number").

- Auf der nächsten Seite einfach mit „**Start exploring the console**“ ohne etwas auszufüllen auf die nächste Seite wechseln.



### You're all set to...

- Publish your apps and skills across millions of devices in over 236 countries and territories.
- Build for natural voice experiences.
- Start generating revenue in a whole new way with Amazon's unique services and APIs.

## Welcome to Amazon Developer Portal

Tell us about your interests so we can keep you informed about our products and feature updates.

Select the products you're interested in:

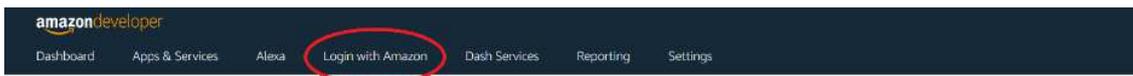
- |   |  |  |
|---|--|--|
| <input type="checkbox"/> Fire TV          | <input type="checkbox"/> Amazon Web Services                     | <input type="checkbox"/> Mobile apps and games |
| <input type="checkbox"/> Twitch           | <input type="checkbox"/> Alexa Voice Services                    | <input type="checkbox"/> Fire Tablets          |
| <input type="checkbox"/> Alexa Skills Kit | <input type="checkbox"/> Dash Replenishment                      | <input type="checkbox"/> Amazon Moments        |
| <input type="checkbox"/> Amazon GameOn    | <input type="checkbox"/> Software and Video games for PC and MAC |  |

Receive product information and updates

Complete your profile now, or finish it later

Add payment information, user roles and other settings to complete your profile. This can be done later through the Settings tab.

- Jetzt müssen wir als erstes ein Sicherheitsprofil anlegen und wählen hierzu „Login with Amazon“ aus:



For developers with live apps in the Japanese marketplace. On October 1, 2019, the consumption tax rate will increase from 8% to 10% for content sold to customers living in Japan. Please find more details [here](#).

### Notifications

All	Critical
No Notifications.	

### Announcements

In-Skill Purchasing (ISP) Now Available in France, Italy, and Spain	Nov 20, 2019	New Developer Tools to Reach More Customers
Announcing New Smart Home Features	Sep 25, 2019	Build Dynamic Gadgets, Games, and Smart Toys w/ Alexa
Build Dialogs with Less Effort, Less Code, and Less Training Data	Jun 5, 2019	In-Skill Purchasing (ISP) Expands to the UK, Germany and Japanese Skill Stores

- Folgendes Fenster erscheint und wir entscheiden uns für das Einrichten:



### Anmeldung mit Amazon

## Sie haben kein "Login bei Amazon" eingerichtet. Jetzt starten.

Anmeldung mit Amazon ermöglicht Benutzern die Anmeldung bei Drittpartei-Websites oder Anwendungen ('Clients') mit ihrem Amazon-Benutzernamen und -Passwort. Clients können den Nutzer auffordern, einige persönliche Informationen aus dem Amazon-Profil freizugeben, einschließlich Name, E-Mail-Adresse und Postleitzahl.

Login mit Amazon-Funktionen sind mit dem Sicherheitsprofil verknüpft, das Sie für Ihre Websites und Apps erstellen. Um zu beginnen, klicken Sie auf **Ein neues Sicherheitsprofil erstellen**.

[Erstellen eines neuen Sicherheitsprofils](#)



## Anmeldung mit Amazon

Verbinden Sie sich sicher mit Apps und Websites

- In diesem Fenster einfach die Felder (**Name, Beschreibung, Hinweis**) ausfüllen. Ich habe als Datenschutzhinweis einfach die AGB Adresse von Amazon genommen... Mit speichern dann abschließen:

- Wir haben nun einen OAuth2 Sicherheitsprofil erstellt nun brauchen wir aus diesem Profil noch die Daten und klicken auf das erstellte Profil.

Sicherheitsprofil Name	OAuth2-Anmeldeaten	Verwalten
smarthomeng	<a href="#">Client-ID und Client-Geheimnis anzeigen</a>	

- Die **Client ID** und das **Client Geheimnis** brauchen wir für später! Diese beiden Schlüssel bitte in eine Textdatei kopieren, dann brauchen wir die später nur von dort aus wieder kopieren.

[Home](#)
[Documentation](#)
[Login with Amazon Console](#)

## Anmeldung mit Amazon

Anmeldung mit Amazon ermöglicht Benutzern die Anmeldung bei Drittpartei-Websites oder Anwendungen (Clients) mit Ihrem Amazon-Benutzernamen und -Passwort. Clients können den Nutzer auffordern, einige persönliche Informationen aus dem Amazon-Profil freizugeben, einschließlich Name, E-Mail-Adresse und Postleitzahl. Wählen Sie zunächst ein vorhandenes Sicherheitsprofil oder erstellen Sie ein neues Sicherheitsprofil. [Weitere Informationen](#)

[Erstellen eines neuen Sicherheitsprofils](#)

✔ Login bei Amazon erfolgreich aktiviert für Sicherheitsprofil. Klicken Sie [hier](#) zum Verwalten des Sicherheitsprofils.

Login mit Amazon-Konfigurationen

Sicherheitsprofil Name	OAuth2-Anmeldedaten	Verwalten
smarthomeng	Client ID: amzn1.application-oa2-client: [REDACTED] Client-Geheimnis: f3df1ad6028d [REDACTED]	<a href="#">Verwalten</a>

- Das Sicherheitsprofil OAuth2 ist nun erstellt. Und wir wechseln erst einmal in die Einrichtung des AWS Accounts....

## 6a: Alexa Skill anlegen:

Jetzt muss noch der Skill unter <https://developer.amazon.com/alexa/console/ask> mit „Create Skill“ erstellt werden:

[Skills](#)
[Earnings](#)
[Payments](#)
[Hosting](#)

### Alexa Skills

[Create Skill](#)

SKILL NAME	LANGUAGE	TYPE	MODIFIED	STATUS	ACTIONS
 <p><b>Alexa Skills</b></p> <p>Create your first skill or learn more about <a href="#">Alexa Skills Kit</a></p> <p><a href="#">Create Skill</a></p>					

- Wir müssen nun einen **Namen** für den neuen Skill benennen. Wir nehmen „**SmartHomeNG**“. Wählen dann noch das Skill Model aus (**Smart Home**) und als Methode „**Provision your own**“ und erstellen dann den Skill mit noch einem Klick auf „**Create skill**“.

**Skill name**

11/50 characters

**Default language**

More languages can be added to your skill after creation

### 1. Choose a model to add to your skill

There are many ways to start building a skill. You can design your own custom model or start with a pre-built model. Pre-built models are interaction models that contain a package of intents and utterances that you can add to your skill.

<p><b>Custom</b></p> <p>Design a unique experience for your users. A custom model enables you to create all of your skill's interactions.</p>	<p><b>Flash Briefing</b></p> <p>Give users control of their news feed. This pre-built model lets users control what updates they listen to.</p> <p>"Alexa, was ist in den Nachrichten?"</p>	<p><b>Smart Home</b></p> <p>Give users control of their smart home devices. This pre-built model lets users turn off the lights and other devices without getting up.</p> <p>"Alexa, schalte das Licht ein"</p>
---	---	---

### 2. Choose a method to host your skill's backend resources

You can provision your own backend resources or you can have Alexa host them for you. If you decide to have Alexa host your skill, you'll get a to our code editor, which will allow you to deploy code directly to AWS Lambda from the developer console.

Provision your own

Hier beißt sich die Katze etwas in den Schwanz. Denn wir benötigen die **Endpoint** Angabe aus der fertigen Lambda-Funktion die es ja jetzt noch nicht gibt und bei der Erstellung der Lambda-Funktion benötigen wir die **Skill ID** die hier ja erst erstellt wird. Wir müssen nun also zwangsläufig etwas zwischen den beiden (AWS-Lambda / Skill erstellen) Einrichtungsvorgängen hin und herspringen.

- Wir wählen als erstes hier die Payload Version: **v3 (preferred)**
- Hier ist nun die korrekte geographische Region als **Default Endpoint** über die Lambda Funktion zu wählen. „eu-west“ wird dann später über die Lambda Endpoint Angabe eingestellt dort kopiert und hier eingefügt (Haben wir zum jetzigen Zeitpunkt noch nicht). Und es muss noch **Europe/India** angehakt sein.
- Das was unter „**Your skill ID**“ steht, bitte auch in die Zwischenablage **oder** die Textdatei kopieren.

**Das benötigen wir dann später zur Erstellung der Lambda-Funktion.**

alex developer console

< Your Skills SmartHomeNG Build Code Test Distribution Certification Analytics Feedback

German (DE)

**SMART HOME**

ACCOUNT LINKING

PERMISSIONS

1. Payload version <sup>ⓘ</sup>

v3 (preferred)

v2 (legacy-deprecated; please select v3)

2. Smart Home service endpoint

AWS Lambda ARN <sup>ⓘ</sup>

Your Skill ID: amzn1.ask.skill.adf1d9[REDACTED] [Copy to clipboard](#)

Default endpoint\* <sup>ⓘ</sup>: arn:aws:lambda:location<aws\_account\_id>:function:<lambda\_name>

Pick a geographical region that is closest to your target customers and setup geographic specific endpoints:

North America <sup>ⓘ</sup>: arn:aws:lambda:us-east-1:<aws\_account\_id>:function:<lambda\_name>

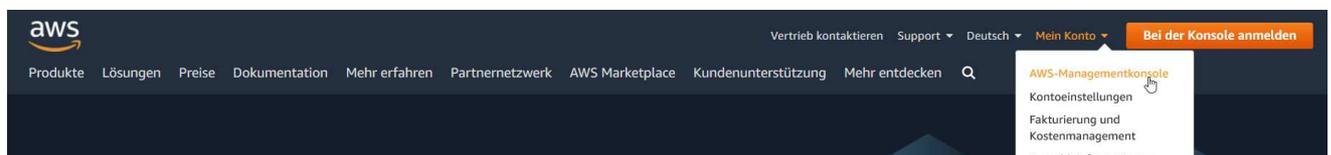
Europe, India <sup>ⓘ</sup>: arn:aws:lambda:eu-west-1:<aws\_account\_id>:function:<lambda\_name>

Far East <sup>ⓘ</sup>: arn:aws:lambda:location<aws\_account\_id>:function:<lambda\_name>

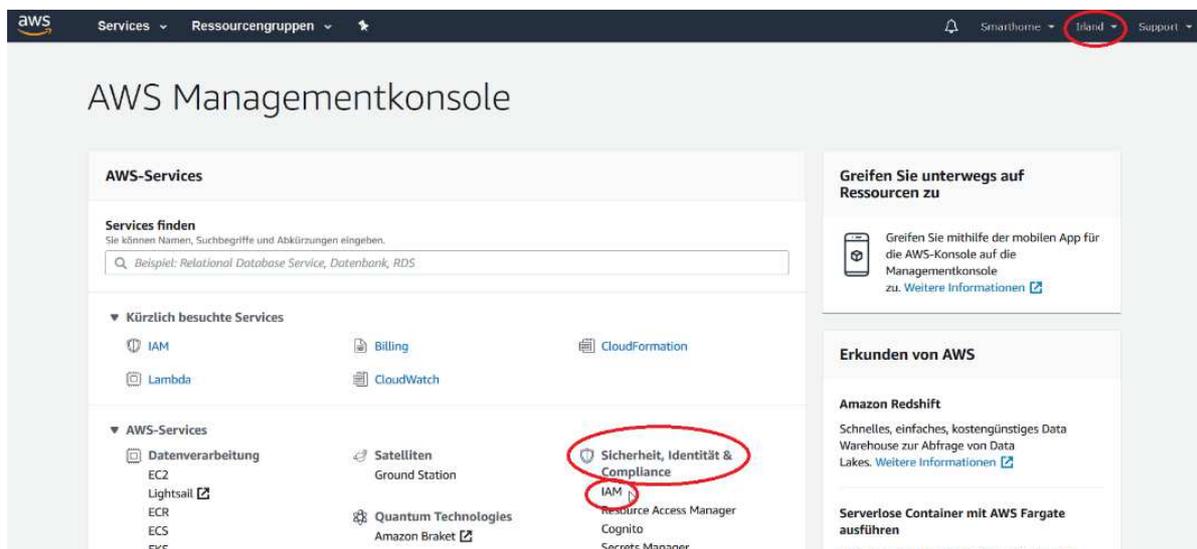
- Da wir nun ohne die Endpoint Angabe aus der AWS Lambda-Funktion nicht weiterkommen (Wir können ohne die Angabe nicht speichern), lassen wir die Seite offen und machen erst einmal auf einem neuen Tab im Browser mit dem AWS Account weiter....

## 7: Einen Amazon AWS Account erstellen und Lambda Service einrichten:

- Nun muss über <https://aws.amazon.com> eine sog. Lambda-Funktion angelegt werden.
- **Bitte erst ein neues Konto anlegen. Der Kontoname muss nicht der Selbe sein wie beim Alexa development Account. Bitte die gesamte Erstellung des Account durchlaufen.**
- Anschließend an dem neu erstellten AWS Account oben recht bei Konto die Option Managementkonsole auswählen.



- **Als erstes müssen wir nun eine neue Rolle (lambda\_basic\_execution) erstellen...**
- Als erstes oben rechts das Land „Irland“ auswählen.
- Dann als Service „Sicherheit Identität & Compliance“ und dort den IAM wählen



- Hier auf der linken Seite „Rolle“ auswählen:

aws Services Ressourcengruppen

Identity and Access Management (IAM)

Dashboard

Zugriffsverwaltung

- Gruppen
- Benutzer
- Rollen**
- Richtlinien
- Identitätsanbieter
- Kontoeinstellungen

Willkommen bei Identity and Access Management (IAM)

Anmelde-Link für IAM-Benutzer:

IAM-Ressourcen

Benutzer: 0 Rollen: 5

Gruppen: 0 Identitätsanbieter: 0

Vom Kunden verwaltete Richtlinien: 2

Sicherheitsstatus

Stammzugriffsschlüssel löschen

- Und eine neue Rolle erstellen:

aws Services Ressourcengruppen

Identity and Access Management (IAM)

Dashboard

Zugriffsverwaltung

- Gruppen
- Benutzer
- Rollen**
- Richtlinien
- Identitätsanbieter
- Kontoeinstellungen

Zugriffsberichte

- Zugriffsanalysator
- Archivregeln
- Analysatordetails

Bericht mit Anmeldeinformationen

Organisationsaktivität

Rollen

Was sind IAM-Rollen?

IAM-Rollen sind eine sichere Methode zum Erteilen von Berechtigungen für Entitäten, denen Sie vertrauen. Nachfolgend finden Sie Beispiele von Entitäten:

- IAM-Benutzer eines anderen Kontos
- Auf einer EC2-Instance ausgeführter Anwendungscode, der Aktionen auf AWS-Ressourcen durchführen muss
- Ein AWS-Service, der Ressourcen in Ihrem Konto handhaben muss, um Funktionen bereitzustellen
- Benutzer aus einem Unternehmensverzeichnis, die den Identitätsverbund mit SAML verwenden

IAM-Rollen erteilen für kurze Zeit gültige Schlüssel, was eine sicherere Methode zur Zugriffsgewährung darstellt.

Weitere Ressourcen:

- Häufig gestellte Fragen zu IAM-Rollen
- IAM-Rolldokumentation
- Anleitung: Kontoübergreifenden Zugriff einrichten
- Gängige Szenarien für Rollen

Rolle erstellen Rolle löschen

- Hier „AWS-Service“ wählen und auf „Lambda“ klicken und unten rechts auf „Weiter: Berechtigungen“ klicken.

aws Services Ressourcengruppen Smarthome Global

Rolle erstellen

Typ der vertrauenswürdigen Entität auswählen

- AWS-Service** (EC2, Lambda und andere)
- Anderes AWS-Konto (Gehört Ihnen oder einem Drittanbieter)
- Web-Identität (Cognito oder ein beliebiges OpenID-Anbieter)
- SAML 2.0-Verbund (Für Firmenverzeichnisse)

AWS-Service: **Wählen Sie den Service aus, der diese Rolle verwendet.**

EC2: Allows EC2 instances to call AWS services on your behalf.

**Lambda**: Allows Lambda functions to call AWS services on your behalf.

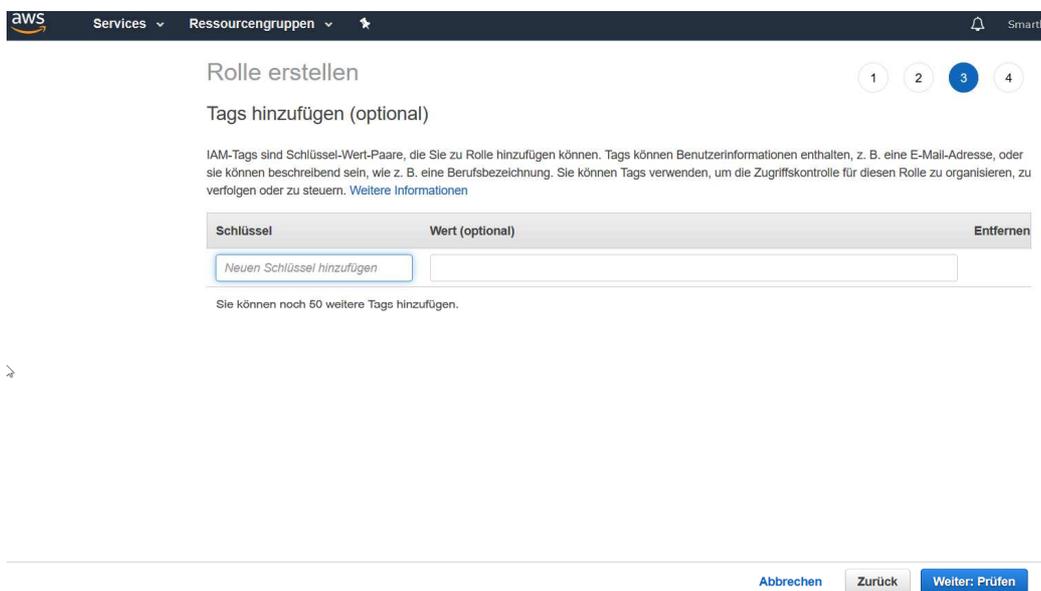
API Gateway CodeDeploy EMR KMS RoboMaker

AWS Backup CodeGuru ElastiCache Kinesis S3

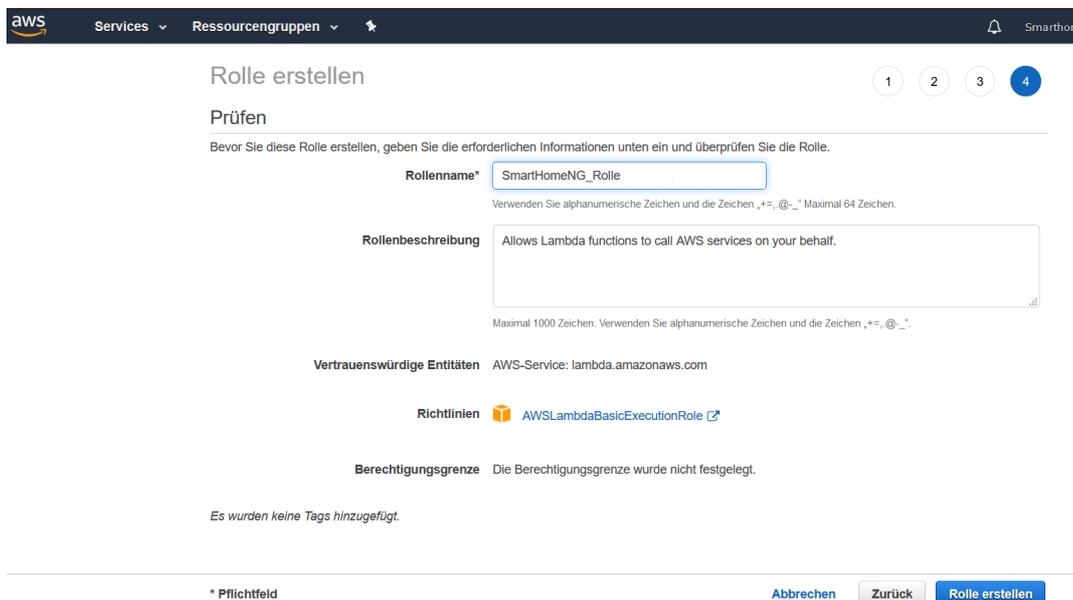
- In das Suchfeld nun „**basic**“ schreiben und bei den Suchergebnissen „**AWSLambdaBasicExecutionRole**“ anhaken.
- Anschließend unten rechts mit „**weiter : Tags**“ weitergehen.



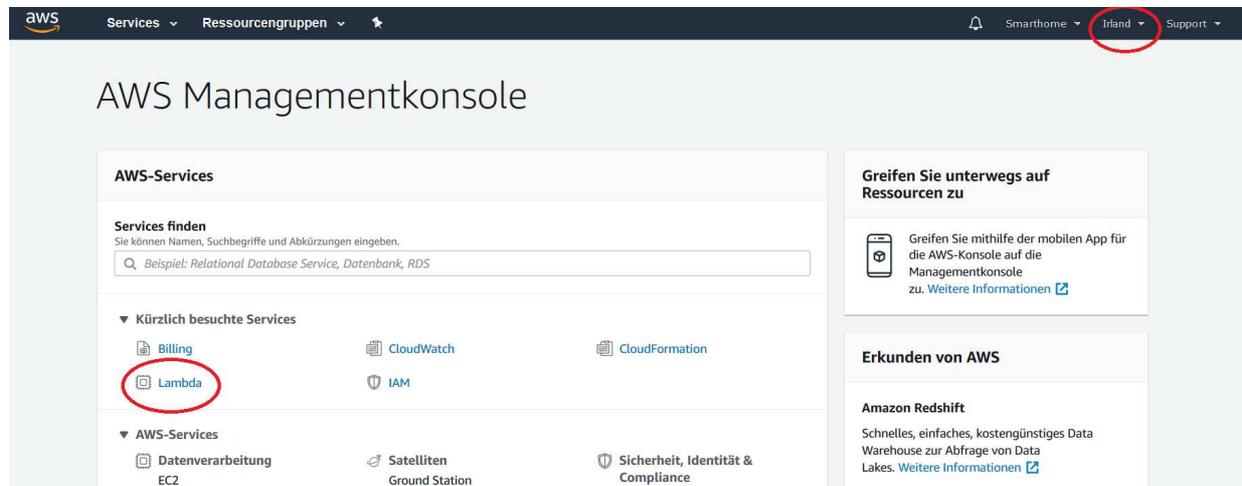
- Die Tags überspringen wir mit dem Button unten rechts „**weiter : Prüfen**“



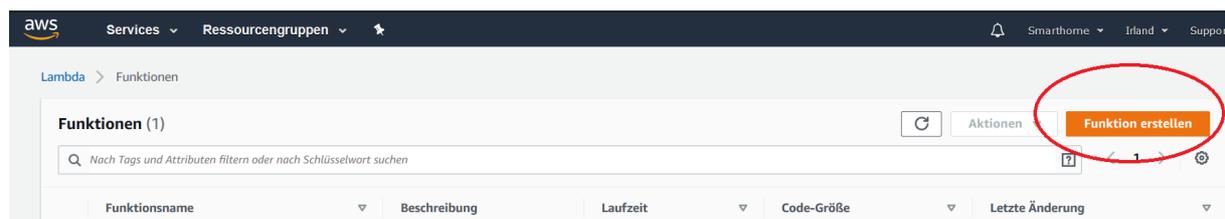
- Jetzt vergeben wir noch einen Namen für die neue Rolle: „**SmartHomeNG\_Rolle**“ und schließen den Vorgang mit „**Rolle erstellen**“ ab.



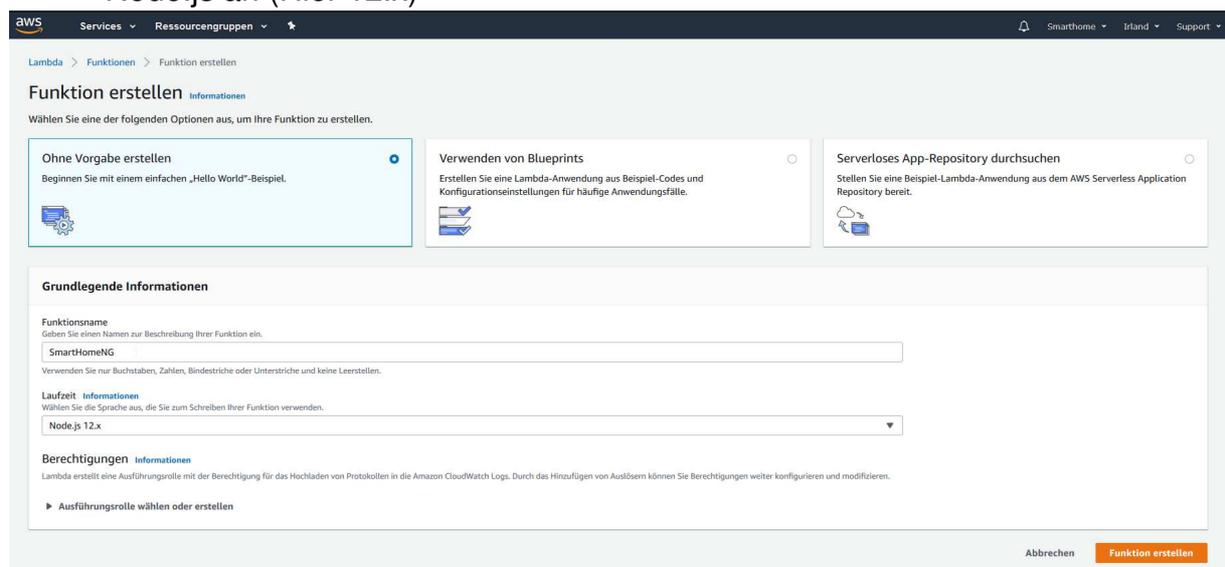
- Wir gehen nun zurück in die Managementkonsole.
- Und den AWS-Service „**Lambda**“ wählen.



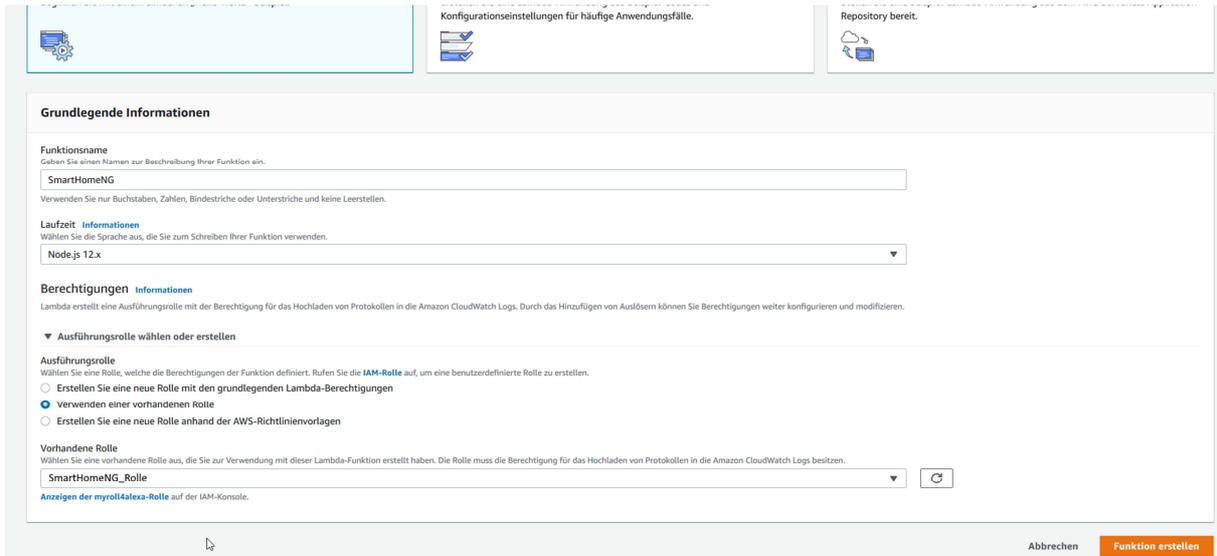
- Im nächsten Fenster erstellen wir eine neue Lambda Funktion mit dem Button „**Funktion erstellen**“



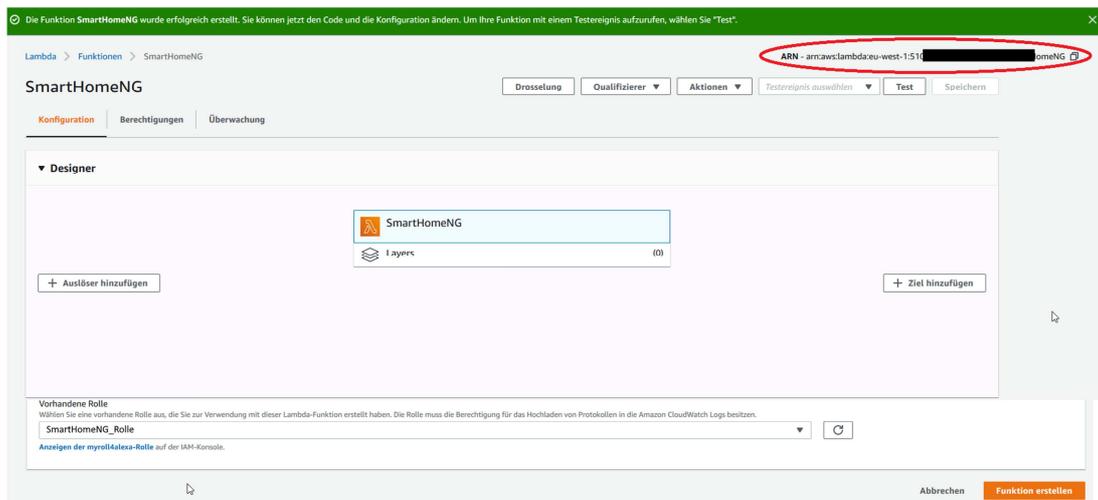
- Wir wählen hier „**Ohne Vorgabe erstellen**“ und vergeben der neuen Funktion einen Namen (**SmartHomeNG**). Bei „**Laufzeit**“ geben wir den aktuellen Node.js an (Hier 12.x)



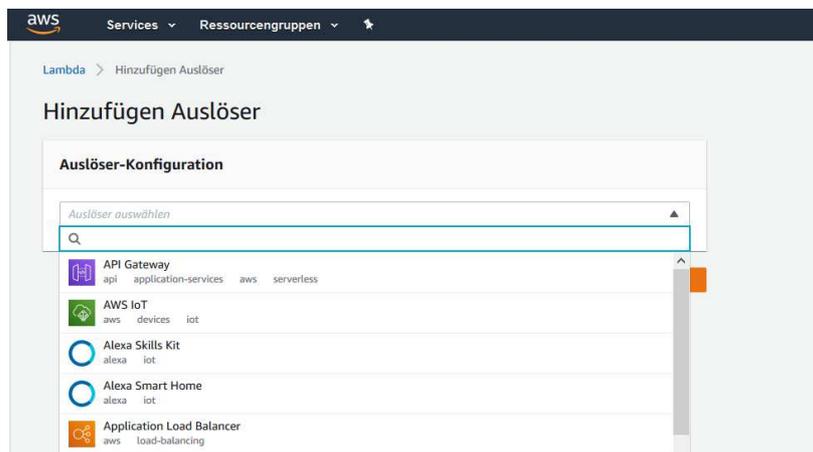
- Anschließend noch unten das Menü „**Ausführungsrolle wählen oder erstellen**“ aufklappen und unsere vorher erstellte Rolle auswählen (**SmartHomeNG\_Rolle**).
- Anschließend mit dem Button unten rechts die **Funktion erstellen**.



- Nun öffnet sich das Fenster um die neue Funktion zu konfigurieren.
- Hier notieren wir als erstes die **ARN Nummer** oben rechts. Das ist dann die Nummer die wir als **Endpoint** in dem **Alexa Skill** eintragen werden. Also auf das Kopiersymbol neben der Nummer klicken und irgendwo hin kopieren (Textdatei) für die spätere Nutzung.
- Anschließend klicken wir auf „**Auslöser hinzufügen**“



- Als Auslöser wählen wir hier **Alexa Smart Home**.



- Und fügen dann die **Anwendungs-ID** (Your Skill ID) aus dem Alexa Skill hier ein. Die ID fängt mit „**Amzn1.ask.skill...**“ an.
- Dann noch den Auslöser aktivieren (Haken) und mit „**Hinzufügen**“ hinzufügen.

The screenshot shows the AWS Lambda console interface for adding a trigger. The main heading is 'Hinzufügen Auslöser'. Under 'Auslöser-Konfiguration', the trigger type is set to 'Alexa Smart Home'. There is a text input field for the 'Anwendungs-ID'. A checkbox for 'Auslöser aktivieren' is checked. The 'Hinzufügen' button is highlighted in orange.

- Dann sollte das so aussehen:

The screenshot shows the SmartHomeNG configuration interface. The 'Designer' section is active, showing a central 'SmartHomeNG' block with a 'Layers' section below it. An 'Alexa Smart Home' trigger block is being added to the left. The interface includes navigation tabs like 'Konfiguration', 'Berechtigungen', and 'Überwachung', and various action buttons like 'Drosselung', 'Qualifizierer', 'Aktionen', 'Test', and 'Speichern'.

- Wir drücken hier nun auf den Block in der Mitte (SmartHomeNG).
- **!! Falls ein Demo-Code mit „Hello from Lambda“ vorhanden ist, diesen auf jeden Fall entfernen !!**
- Als Funktionscode (index.js) kopieren wir folgenden Code in das Fenster:

```

/*
You need to specify the following environmental variables in the lambda function:
- SMARTHOME_HOST
    foobar.dyndns.tld
- SMARTHOME_PORT
    443 - endpoint must be https enabled!
- SMARTHOME_PATH
    '/'
- SMARTHOME_AUTH
    'user:password'
*/
exports.handler = function(event, context, callback) {
    var data = JSON.stringify(event)

    var options = {
        hostname: process.env.SMARTHOME_HOST,
        port: process.env.SMARTHOME_PORT,
        path: process.env.SMARTHOME_PATH,
        method: 'POST',
        auth: process.env.SMARTHOME_AUTH,
        headers: {
            'Content-Type': 'application/json',
            'Content-Length': Buffer.byteLength(data)
        }
    };

    var https = require('https');
    var req = https.request(options, (res) => {
        console.log(`HTTP ${res.statusCode}`);
        res.setEncoding('utf8');

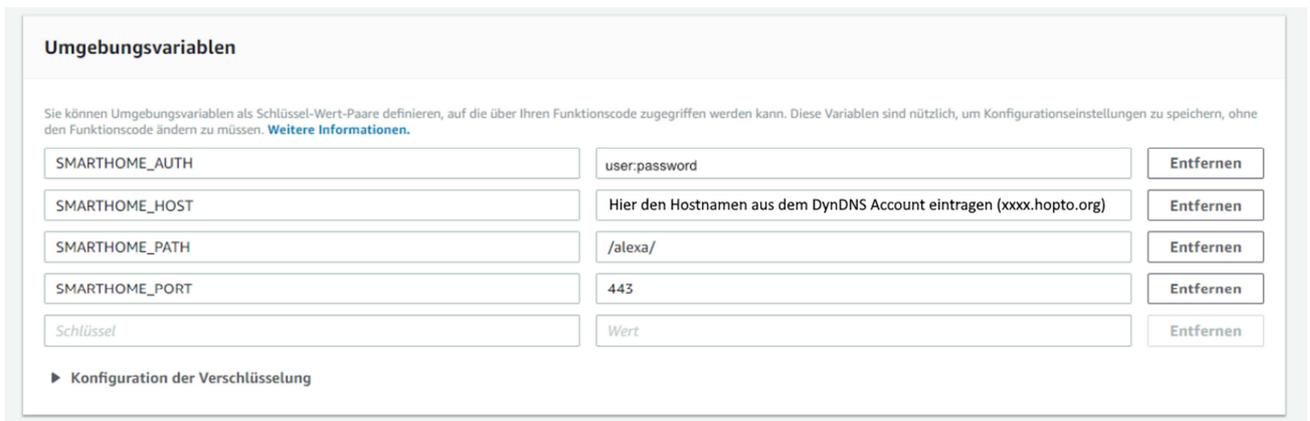
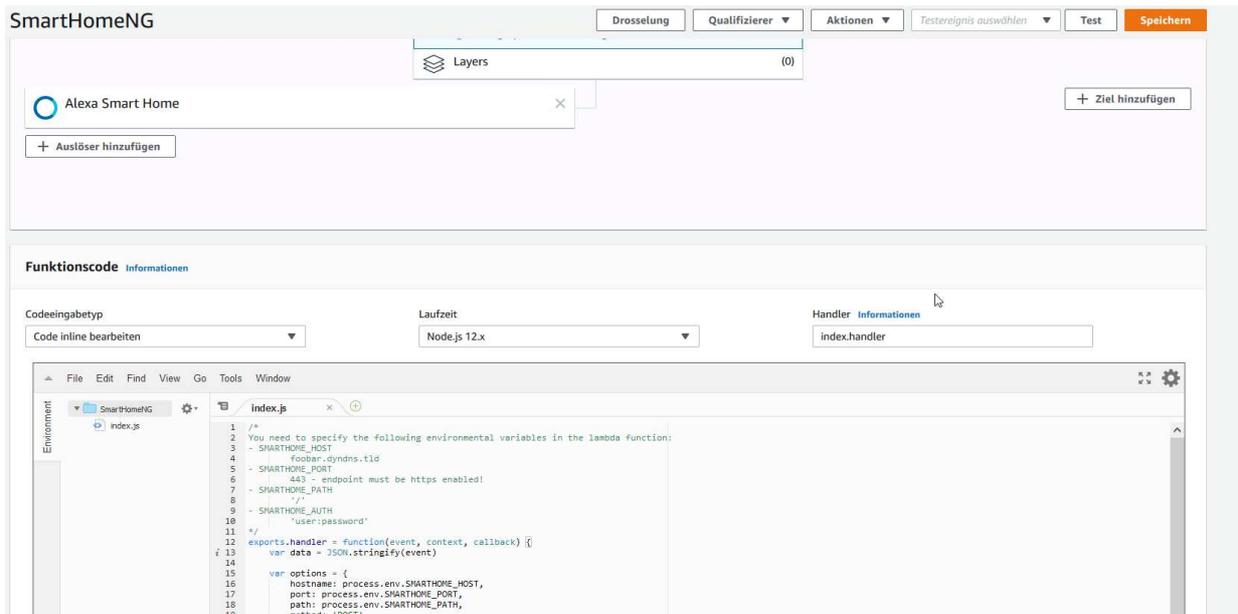
        var responseData = "";
        res.on('data', (dataChunk) => {
            responseData += dataChunk
        });
        res.on('end', () => {
            console.log('raw response:', responseData)

            var response = JSON.parse(responseData);
            if (res.statusCode === 200) {
                console.info('OK', JSON.stringify(response))
                callback(null, response);
            } else {
                console.error('Failed', JSON.stringify(response))
                callback('DependentServiceUnavailableError');
            }
        });
    });
    req.on('error', (e) => {
        console.error('request failed', e);
        callback(e);
    });

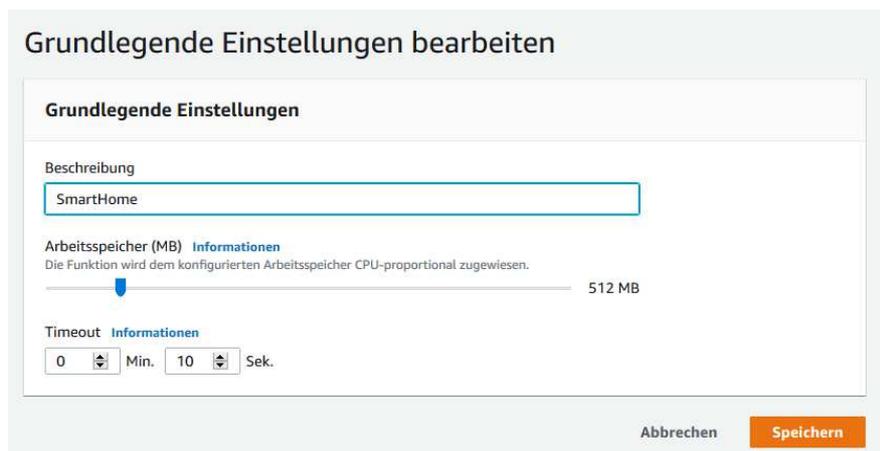
    console.log('requesting', data)
    req.write(data);
    req.end();
}

```

- Das ganze sollte dann so aussehen:



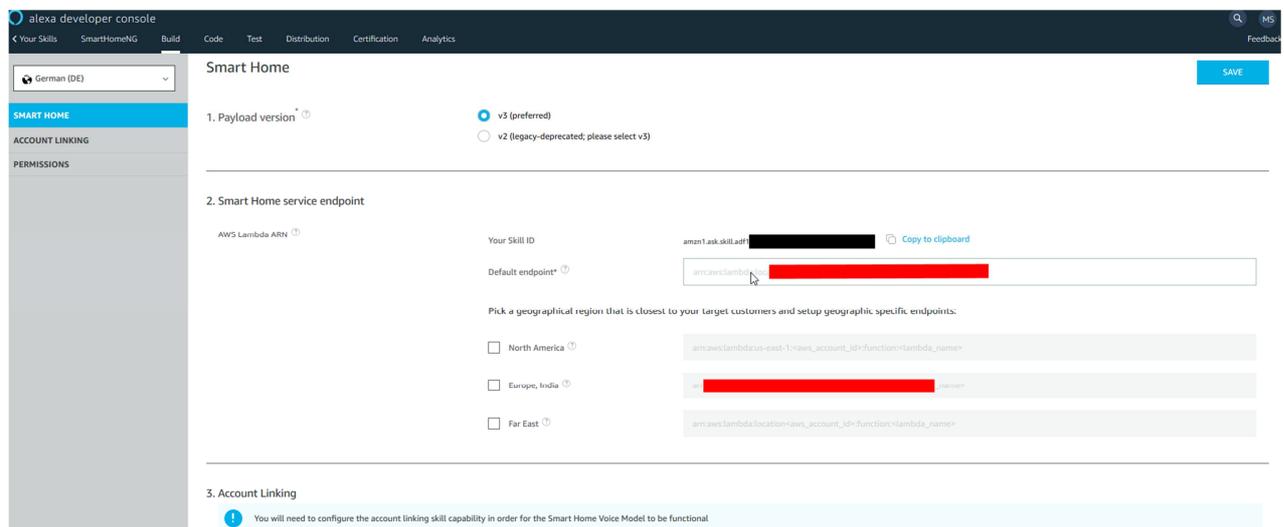
- Die **Grundlegenden Einstellungen** setzen wir folgendermaßen:



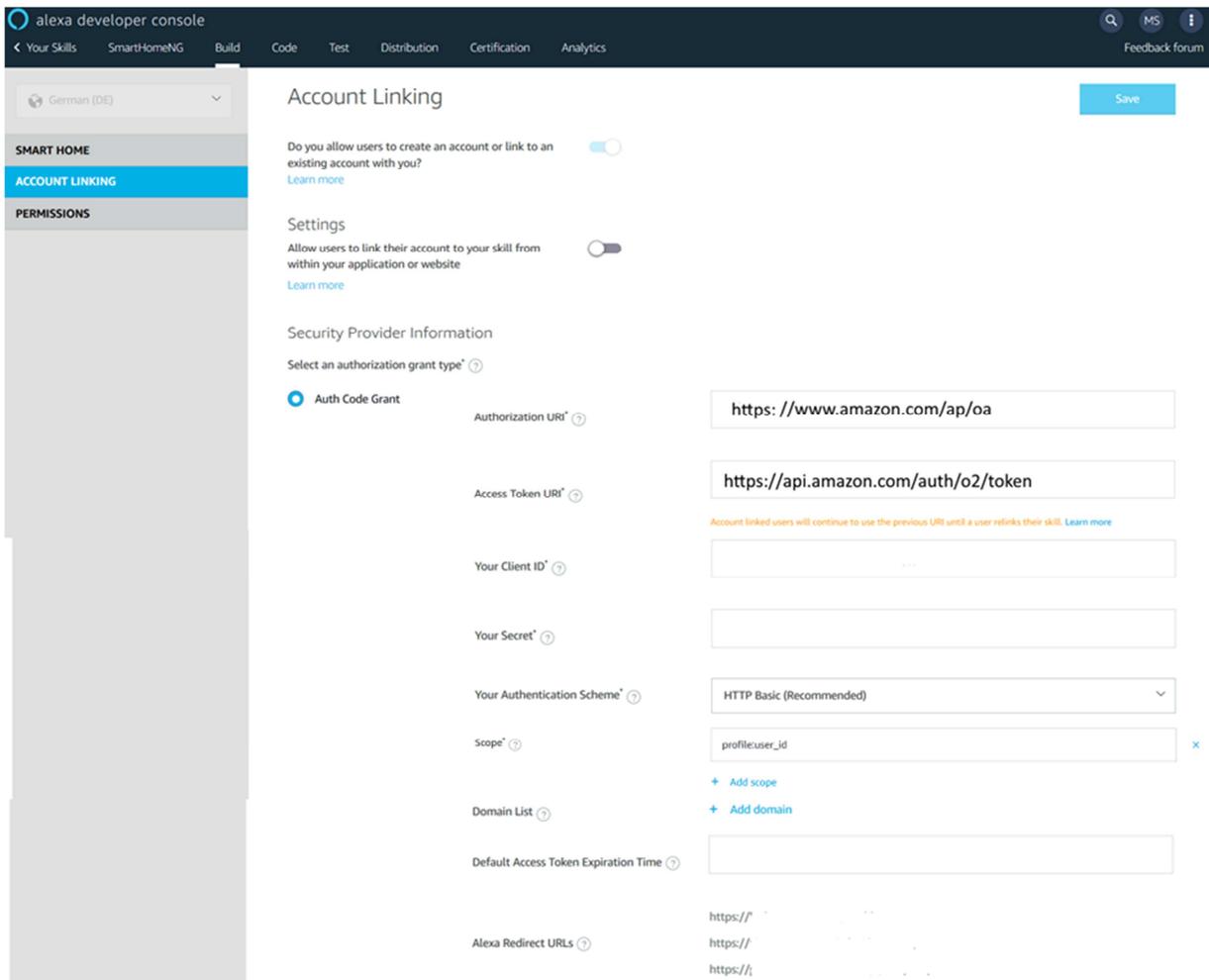
- Anschließend ganz oben rechts mit „**Speichern**“ die komplette Lambda Funktion speichern.
- Damit ist die Lambda Funktion erstellt!

## 6b Alexa Skill:

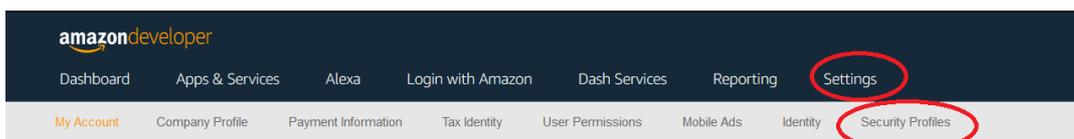
- Jetzt müssen wir noch die **ARN Nummer** (Fäng so an: arn:aws:lambda:eu-west-1...) im **Alexa Skill** als **Endpoint** an den roten Stellen einfügen und **Europe/India** anhaken/aktivieren:



- Jetzt noch auf „**Setup Account Linking**“ klicken und dort die restlichen Daten eintragen.
- Hier benötigen wir nun die **Client ID** und das **Client Geheimnis** aus **Punkt 5**.
- Die Client ID bei „**Your Client ID**“ eintragen.
- Das Client Geheimnis bei „**Your Secret**“ eintragen.
- „**Authorization URI**“ bitte mit <https://www.amazon.com/ap/oa> füllen.
- „**Access Token URI**“ bitte mit <https://api.amazon.com/auth/o2/token> füllen.
- Es wurde berichtet, dass ein „Scope“ nötig sei. Sicherheitshalber also hier noch ein **Scope** mit dem + hinzufügen ( profile:user\_id ). Sieht dann aus wie im Bild unten.
- Die drei „**Alexa Redirect URLs**“, ganz unten, aufschreiben oder in eine Text Datei kopieren. Diese benötigen wir dann noch in dem letzten Schritt.

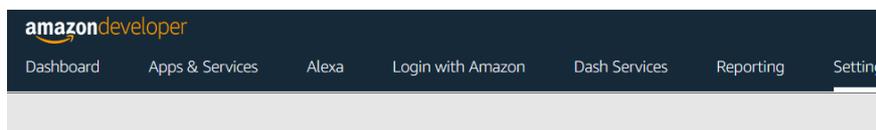


- Am Schluss noch die drei URLs im Alexa Sicherheitsprofil hier eintragen:



My Account

- Dann auf das **Sicherheitsprofil** klicken (hier **smarthomeng**)



## Sicherheits-Profilverwaltung

### Sicherheitsprofile

Um bestimmte Funktionen zu verwenden, muss sich Ihre App sicher bei Amazon identifizieren. Verwenden Sie dieses

 smarthomeng

- Und unter „**Webeinstellungen**“ -> „**Zulässige Rückleitungs URLs**“ die drei URLs eintragen:

The screenshot shows the Amazon Developer console interface. At the top, there's a navigation bar with 'amazon developer' logo and various menu items like 'Dashboard', 'Apps & Services', 'Alexa', 'Login with Amazon', 'Dash Services', 'Reporting', and 'Settings'. Below this, the main content area is titled 'Sicherheits-Profilverwaltung'. Underneath, there's a sub-header 'smarthomeng - Sicherheitsprofil'. A series of tabs are visible: 'Allgemeines', 'Webseiten', 'Android/Kindle-Einstellungen', 'iOS Einstellungen', and 'Fernsehergeräte und andere Geräte-Einstellungen'. The 'Webseiten' tab is active. Below the tabs, there's a paragraph of text explaining the requirements for using Amazon login with a website. Then, there are fields for 'Client-ID' (partially obscured) and 'Client-Geheimnis' with a 'Geheimnis anzeigen' button. Below that, there's a section for 'Zulässige Originale' and 'Zulässige Rückleitungs-URLs'. At the bottom right, there's a 'Bearbeiten' button.

## 8: In der Alexa App den neuen Skill aktivieren und neue Geräte suchen

- In der Alexa APP, im Menü (links oben) den Eintrag „**Skills und Spiele**“ wählen.
- Dort gibt oben einen Reiter (ganz oben) „**Ihre Skills**“. Diesen auswählen.
- Dann gibt es im oberen Bereich drei Optionen (Die aktivierten Skills, Blueprint und nun auch **Entwickler**). Unter Entwickler sollte nun der eigene Skill (**Alexa SmarthomeNG**) erscheinen.
- Diesen **Skill** nun anklicken und anschließend **aktivieren** und **verbinden**.
- Anschließend in die Rubrik **Geräte** (Ganz unten in der Taskleiste) gehen.
- Mit dem + Zeichen (oben rechts) Geräte hinzufügen.
- Dann auf die Option „**Gerät hinzufügen**“ klicken.
- In der Auswahlliste ganz nach unten scrollen und „**Sonstige**„ wählen.
- Anschließend auf „**GERÄTE SUCHEN**“ drücken.
- Die neuen Geräte werden gesucht (Kann 45 Sekunden dauern).
- Anschließend sollte die neue Lampe (Wohnzimmer Licht) aus der Item.yaml gefunden worden sein.

### Sicherheit prüfen:

Auf der Seite <https://www.ssllabs.com/ssltest/index.html> kann man den DynDNS Hostnamen eingeben und prüfen lassen ob alle Verbindungen sicher konfiguriert sind.

**Hier sollte als Ergebnis A+ stehen.**