

# Integración Victron Energy con Node-RED y Amazon Alexa

Este documento describe la integración de un sistema **Victron (Venus OS / Cerbo GX)** con **Amazon Alexa** mediante un túnel de Cloudflare y Node-RED.

**Requisito Previo de Acceso:** Para ejecutar estos pasos, debes habilitar el acceso total en tu dispositivo Victron:

1. Ve a **Settings > General > Access Level**.
2. Cambia el nivel a **Superuser**. El super-usuario se obtiene manteniendo el botón de usuario pulsado por unos segundos hasta que cambien.
3. En el mismo menú, activa la opción **SSH on LAN**. Sin esto, la terminal no permitirá la conexión, actualiza a una clave propia para poder acceder por el terminal.

**Nota de Almacenamiento:** Todo el software (cloudflared), scripts y logs deben residir en una **Tarjeta de Memoria SD formateada en FAT32**, montada usualmente en la ruta /media/TU-TARJETA-SD/. Los comandos se ejecutan mediante una **Terminal (SSH)**.

Todos los comando y programaciones son factibles de copiar y pegar para facilitar la integración, luego solo deberá cambiar ciertos parámetros en los nodos de nodered según sus equipos y Baterías.

## 1. Configuración del Túnel en Cloudflare

1. En el Dashboard de Cloudflare, ve a **Zero Trust > Networks > Tunnels**.
2. Crea un nuevo túnel (ej. "victron").
3. En la pestaña **Public Hostname**:
  - **Subdomain:** victron
  - **Domain:** Tu dominio registrado.
  - **Service:** http://localhost:1880
4. Copia el **Token** proporcionado para el script de inicio.

## 2. Configuración en el Equipo Local (Terminal SSH)

Crea el script de control en la tarjeta SD para asegurar que el sistema no se sature con procesos duplicados.

**Archivo:** /media/mmcb1k0p1/start\_tunnel.sh

**Comando para crear el archivo:**

```
printf '#!/bin/sh

if ps | grep -v grep | grep "cloudflared tunnel" > /dev/null; then
    exit 0
fi
echo "nameserver 1.1.1.1" > /etc/resolv.conf
echo "nameserver 8.8.8.8" >> /etc/resolv.conf

for i in 1 2 3 4 5 6; do
```

```
if ping -c 1 1.1.1.1 > /dev/null 2>&1; then
    break
fi
sleep 5
done
```

```
nohup /run/media/mmcbk0p1/cloudflared tunnel --protocol http2 --heartbeat-
interval 10s --heartbeat-count 3 run --token <TU_TOKEN_AQUI> > /run/media/
mmcbk0p1/tunnel.log 2>&1 &
' > /run/media/mmcbk0p1/start_tunnel.sh
```

**Permisos de ejecución:** `chmod +x /run/media/mmcbk0p1/start_tunnel.sh`

Automatización tras Reinicio

Para que el túnel sea "indestructible", usaremos el crontab. La diferencia aquí es que el script ahora es inteligente: si el túnel está bien, no hace nada; si se cayó, lo levanta.

Ejecuta `crontab -e` y añade estas líneas al final:

```
@reboot /bin/sh /media/mmcbk0p1/start_tunnel.sh
* * * * * /bin/sh /media/mmcbk0p1/start_tunnel.sh
```

### 3. Configuración en Amazon Alexa (Developer Console)

Accede a [developer.amazon.com](https://developer.amazon.com) para configurar la Skill:

#### A. Crear la Skill

1. **Create Skill** > Nombre: Victron Casa.
2. **Default Language:** Selecciona el idioma de tu preferencia (ej. Spanish (ES)).
3. **Model:** Selecciona Custom. **Hosting:** Selecciona Provision your own.

#### B. Interaction Model (Intents)

Configura las frases de activación:

- **BaterialIntent:** "¿cuál es el estado de la batería?", "batería", "porcentaje de carga".
- **SolarIntent:** "¿cuánto producen los paneles?", "producción solar", "vatios solares".
- *Importante:* Haz clic en "**Build Model**" tras guardar.

#### C. Endpoint

1. **Endpoint** > Selecciona **HTTPS**.
2. **Default Region:** `https://tu-dominio.com/api/alexa` (URL de Cloudflare).
3. **SSL Certificate:** Selecciona: "*My endpoint is a sub-domain that has a wildcard certificate...*".

## 4. Lógica en Node-RED

Importa este JSON. Puedes personalizar los textos de respuesta de alexa en el nodo **"Generar Respuesta Inteligente"**.

```
[{"id":"3240ada185f2413b","type":"tab","label":"Alexa Victron"},
{"id":"f27a94492211a6c9","type":"function","z":"3240ada185f2413b","name":"Generar Respuesta Inteligente","func":"const soc = flow.get('soc') || \"no disponible\";\nconst pv = flow.get('pv') || 0;\nconst intentName = (msg.payload.request && msg.payload.request.intent) ? msg.payload.request.intent.name : \"LaunchRequest\";\nlet speak = \"\";\nswitch (intentName) {\n  case \"BateriaIntent\":\n    speak = `El nivel de carga de la batería es del ${soc} por ciento.`;\n    break;\n  case \"SolarIntent\":\n    speak = `Los paneles solares están generando ${pv} vatios en este momento.`;\n    break;\n  default:\n    speak = `Sistema listo. La batería está al ${soc} por ciento y la producción solar es de ${pv} vatios.`;\n}\nmsg.payload = { version: '1.0', response: { outputSpeech: { type: 'PlainText', text: speak }, shouldEndSession: true } };\nreturn msg;","outputs":1,"x":450,"y":320,"wires":[["6baa9a22615b9c3f"]]},
{"id":"d8eacd0659e971f8","type":"http in","z":"3240ada185f2413b","name":"Alexa Endpoint","url":"/api/alexa","method":"post","x":160,"y":300,"wires":[["f27a94492211a6c9"]]},
{"id":"6baa9a22615b9c3f","type":"http response","z":"3240ada185f2413b","name":"HTTP 200","statusCode":"200","x":680,"y":320,"wires":[]},
{"id":"a77968dacbf0c65e","type":"function","z":"3240ada185f2413b","name":"Guardar SOC","func":"flow.set('soc', msg.payload);\nnode.status({fill:'blue',shape:'dot',text:`SOC ${msg.payload}%`});\nreturn null;","outputs":1,"x":540,"y":140,"wires":[[]]},
{"id":"8c6a9162d6321944","type":"function","z":"3240ada185f2413b","name":"Guardar PV","func":"flow.set('pv', msg.payload);\nnode.status({fill:'green',shape:'dot',text:`PV ${msg.payload} W`});\nreturn null;","outputs":1,"x":530,"y":220,"wires":[[]]},
{"id":"5b162252b72956d8","type":"victron-input-battery","z":"3240ada185f2413b","service":"com.victronenergy.battery/512","path":"/Soc","serviceObj":{"service":"com.victronenergy.battery/512","name":"RUiXU RX Battery","communityTag":"battery"},"pathObj":{"path":"/Soc","type":"float","name":"State of charge (%)"},"name":"SOC Batería","onlyChanges":false,"roundValues":"1","rateLimit":0,"outputs":1,"conditionalMode":false,"outputTrue":"true","outputFalse":"false","debounce":"2000","x":330,"y":140,"wires":[["a77968dacbf0c65e"]]},
{"id":"ff40400a00ffbde9","type":"victron-input-solarcharger","z":"3240ada185f2413b","service":"com.victronenergy.solarcharger/279","path":"/Yield/Power","serviceObj":{"service":"com.victronenergy.solarcharger/279","name":"SmartSolar Charger MPPT 150/45 rev3","communityTag":"solarcharger"},"pathObj":{"path":"/Yield/Power","type":"float","name":"PV Power (W)"},"name":"Potencia Solar","onlyChanges":false,"roundValues":"1","rateLimit":0,"outputs":1,"conditionalMode":false,"outputTrue":"true","outputFalse":"false","debounce":"2000","x":340,"y":220,"wires":[["8c6a9162d6321944"]]}]
```

## 5. Diagnóstico (SSH)

- `ps | grep cloudflared` : Verifica si el proceso está activo.
- `tail -f /run/media/mmcblk0p1/tunnel.log` : Monitorea la conexión.

**Descargo de Responsabilidad:** *Este manual ha sido diseñado para facilitar la integración por parte de usuarios con nociones básicas de programación. El uso de esta información y la ejecución de los comandos proporcionados son responsabilidad exclusiva del usuario. No se asume ninguna responsabilidad por daños derivados del hardware, fallos en la seguridad de la red o interrupciones en el servicio por una configuración inadecuada.*