**Background**

We have WeMo smart plugs all over the house, one of which is connected to a cat water fountain. We have two cats, one that likes to drink from the fountain when it’s on, the other when it is off. The WeMo app is wonderful, and so is the integration with Alexa (Amazon’s Echo digital assistant), however there are two challenges I've seen with this approach:

1. The app and digital assistant are a bit cumbersome to use when we sense a cat near the fountain at 3 am (yes, we are crazy about our fuzzbuckets! 😊).
2. Persons with disabilities may find it challenging to operate the WeMo switch via voice or app.

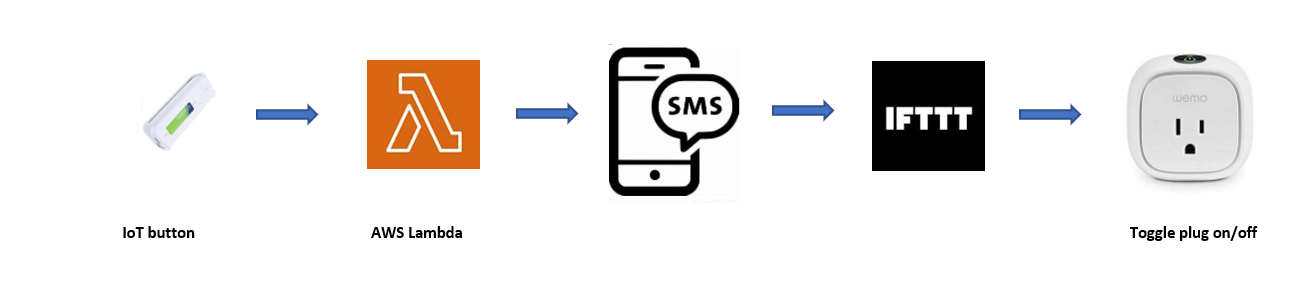
So here’s my attempt to solve for this via Home Automation: toggle the WeMo switch the fountain is plugged into via an IoT button. An IoT button is, well, a button! It’s a Wi-Fi enabled input mechanism that can trigger other actions. Like a doorbell, when someone presses it the chime rings (although the IoT button works without wires).

An IoT button is a “wireless switch” that can turn the connected device on/off from, ostensibly, anywhere in the world. No more issuing a voice command to your digital assistant when you might not have your voice at 3 am or you don’t want to wake up your partner or a sleeping infant. No more opening your phone, opening an app, finding the switch in the list and click it.

**Components**

* [WeMo Smart Plug](https://www.belkin.com/us/p/P-WSP080/) or [light switch](https://www.belkin.com/us/p/P-WLS040/): Internet-enabled electrical outlet.
* [WeMo app](https://play.google.com/store/apps/details?id=com.belkin.wemoandroid&hl=en_US&gl=US): this allows you to control the above.
* [IFTTT account](https://ifttt.com/): If This Then That. This is a way to conditionally synchronize standalone apps.
* [AWS account](https://aws.amazon.com/free/?trk=ps_a134p000006pkldAAA&trkCampaign=acq_paid_search_brand&sc_channel=ps&sc_campaign=acquisition_US&sc_publisher=Bing&sc_category=core&sc_country=US&sc_geo=NAMER&sc_outcome=acq&sc_detail=create%20aws%20account&sc_content=Account_e&sc_matchtype=e&sc_segment=&sc_medium=ACQ-P|PS-BI|Brand|Desktop|SU|AWS|Core|US|EN|Text&s_kwcid=AL!4422!10!71605848472470!71606291145864&s_kwcid=AL!4422!10!71605848472470!71606291145864&ef_id=fa451d632aef16e557807de2d309ba54:G:s&all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=*all&awsf.Free%20Tier%20Categories=*all): Cloud
* [AWS Lambda](https://aws.amazon.com/lambda/): Serverless computing. It’s not \*actually\* serverless, but be that as it may....
* [AWS IoT](https://aws.amazon.com/iot/): Internet connected objects (Internet of Things). Physical objects that are embedded with ‘computer smarts’ (sensors, chips etc.) that they use to communicate with the world.
* [Seeed IoT button](https://www.seeedstudio.com/Seeed-IoT-Button-for-AWS-p-4527.html): this is the IoT button I decided to go with.

**Process**



Click button -> AWS Lambda runs -> SMS is sent to phone -> IFTTT triggers and sends a command to … > WeMo App to toggle the WeMo Smart Plug

**Steps**

* Buy a Seeed IoT button: <https://wiki.seeedstudio.com/SEEED-IOT-BUTTON-FOR-AWS/#operating-instructions>. You'll also need a rechargeable battery since that doesn't come with it. Sorry, I didn’t make up the rules 😊.
* Create an AWS account if you don't have one.
* Create an IFTTT account if you don't have one.
* Launch AWS and access the IoT One-click service and select Claim Your Device. Then enable it.
* Select the AWS IoT One-Click service and select Create a Project. Select a Device Template as All Button Types and the Action is 'Send SMS'. Supply your mobile number and for the message default value, "#WeMo".
* Launch AWS Lambda, and select the default SMS Lambda created to attach to this button. Overwrite the lambda\_function.py with the code provided here. I hard-coded the phone number but you can pass it through the parameters in the AWS Lambda console you set above if you like, by uncommenting the appropriate lines above ‘<YOUR PHONE NUMBER HERE>’ line and removing or commenting out that line.



* Launch IFTTT and Create an Applet. Select 'New SMS Received that matches phone number” and supply the phone number that AWS sends the text message from. You can obtan this by executing the lambda once in test mode.

For the 'Then' part select Wemo and supply your switch type and name and 'Toggle on/off'. Be sure to turn on ‘Get notifications when this Applet is active’ otherwise when the app is in sleep mode on your mobile it will wait for it to wake to toggle the switch.



**Future**

Possible future enhancements:

* Speed up the execution. That may require an IFTTT Pro account. Currently the process takes ~ 20 seconds to complete due to the several handoffs that need to occur. The biggest time-intensive task seems to be the IFTTT trigger, hence the possible enhancement.
* Reporting, to capture usage patterns. Are the cats drinking too much water? Too little? What are their popular times to drink?
* Make the SMS trigger dependent on keyword search of the recipient text message store, rather than the sending phone number. There’s more control and stability that way. The issue is the IFTTT function to search all recipient text for a keyword doesn’t seem to work. [IFTTT has transitioned to using moble notifications for the default account it creates](https://help.ifttt.com/hc/en-us/articles/360023055054-Important-changes-to-the-SMS-service), which could be impacting.
* The IFTTT SMS service is limited to 100 messages per month. Use mobile notifications and/or create a service on the [IFTTT Platform](https://ifttt.com/developers) for higher volume needs.
* Use DeepLens to recognize which cat approaches and act accordingly. I tried this, however there are two challenges:
  + AI was not successful enough in recognizing one cats face over another
  + When it’s dark, big hairballs! (pun intended…cats<->hairballs). No light for Deeplens to work with. Perhaps an IFTTT connection between a motion-sensing light and the Deeplens?

There are many possible applications for IoT buttons. You can set one to order your favorite coffee or to act as a ‘panic button’ or as in the example set forth by Seeed, to quickly garner feedback from a customer. [Let me know](mailto:jpgrieco@outlook.com) what you decide to do!

**Credits**

* Thanks to Dave Mathews for his [Smarthome Control](https://www.hackster.io/ggdm/hello-good-night-smarthome-control-without-a-hub-cbe3f3#code) post which greatly helped setting this up.