

Awesome Automations


Using Webhooks to Automate and Integrate

Tami Miguens





The Power of Cloud Computing


The Power of Cloud Computing





Get a notification when the International Space Station passes over your house

by IFTTT 





works with 

 > 

Create Toggl time entry from Flic buttons

Starting a time can be as simple as pushing a button with the help of Zapier! Connect your time tracking software to [...more](#)

Try It

 > 

Create LastPass accounts for new Namely employees

New hires need to get set up with so many accounts, and it can take hours for an IT or Hiring Manager to do. Save [...more](#)

Try It



Get yourself out of an awkward situation (International)

by IFTTT 



works with 

Getting Started with Webhooks

1. **Introduction** to web services + vocabulary
2. **No code** options for **building integrations**
3. Making **REST API** requests + **JSON** format
4. Experiment with **webhooks** and view the **payload**
5. Building your own webhook custom **endpoint**



Web Services Intro

What is an API?

How apps share data, without your help



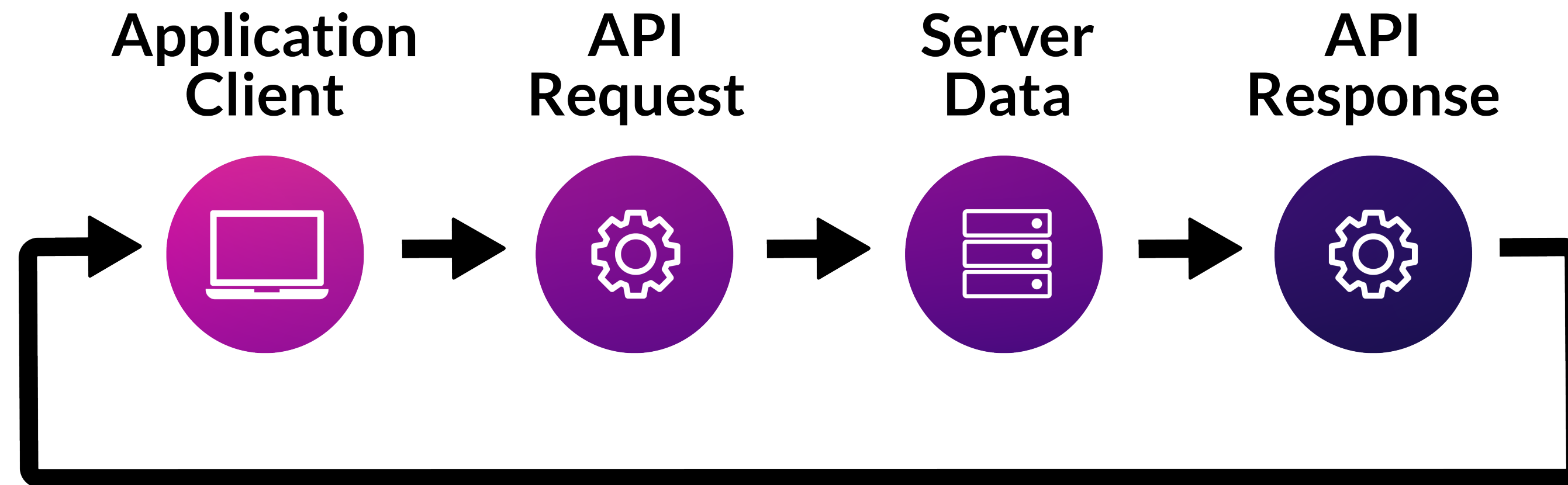
API = Application Programming Interface

- A way for apps to communicate
- Can be open to public or partner-only
- Look for “**Integrations**” or “**Add-ons**” on an app’s website for plug-and-play uses of APIs
- Look for “**API**” on an app’s website to find out if you can build your own integration

Modern APIs use REST

Standard protocols mean everybody can get in on the convo

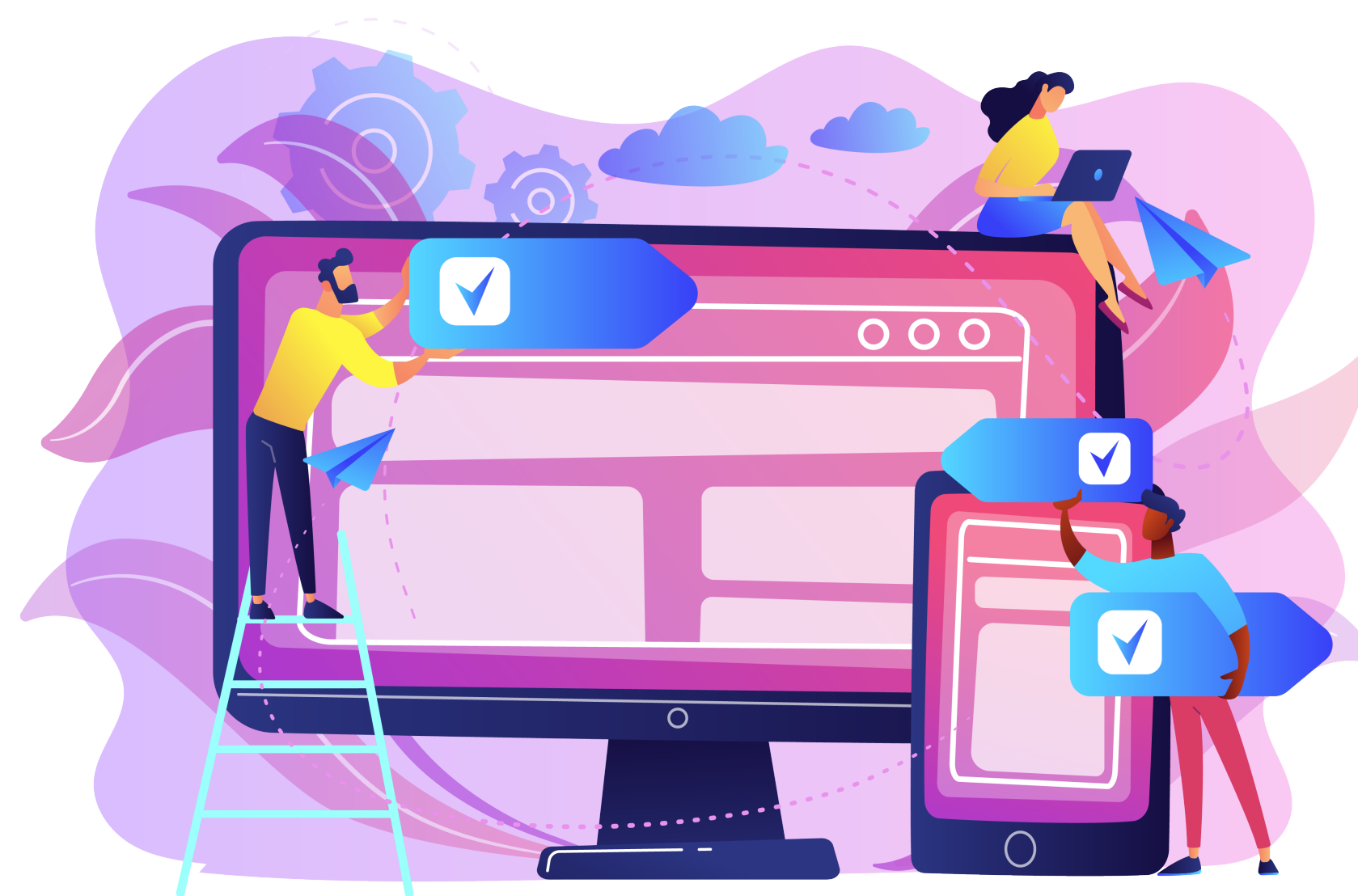
A client sends a request to a server through HTTP, the server (backend) sends back a response



- Get a user's profile
- Upload a receipt
- Update a user's email address
- Get all systems with AppleCare
- Add a new laptop
- Modify a policy

When REST APIs are not enough

When you want to do stuff, not just know stuff

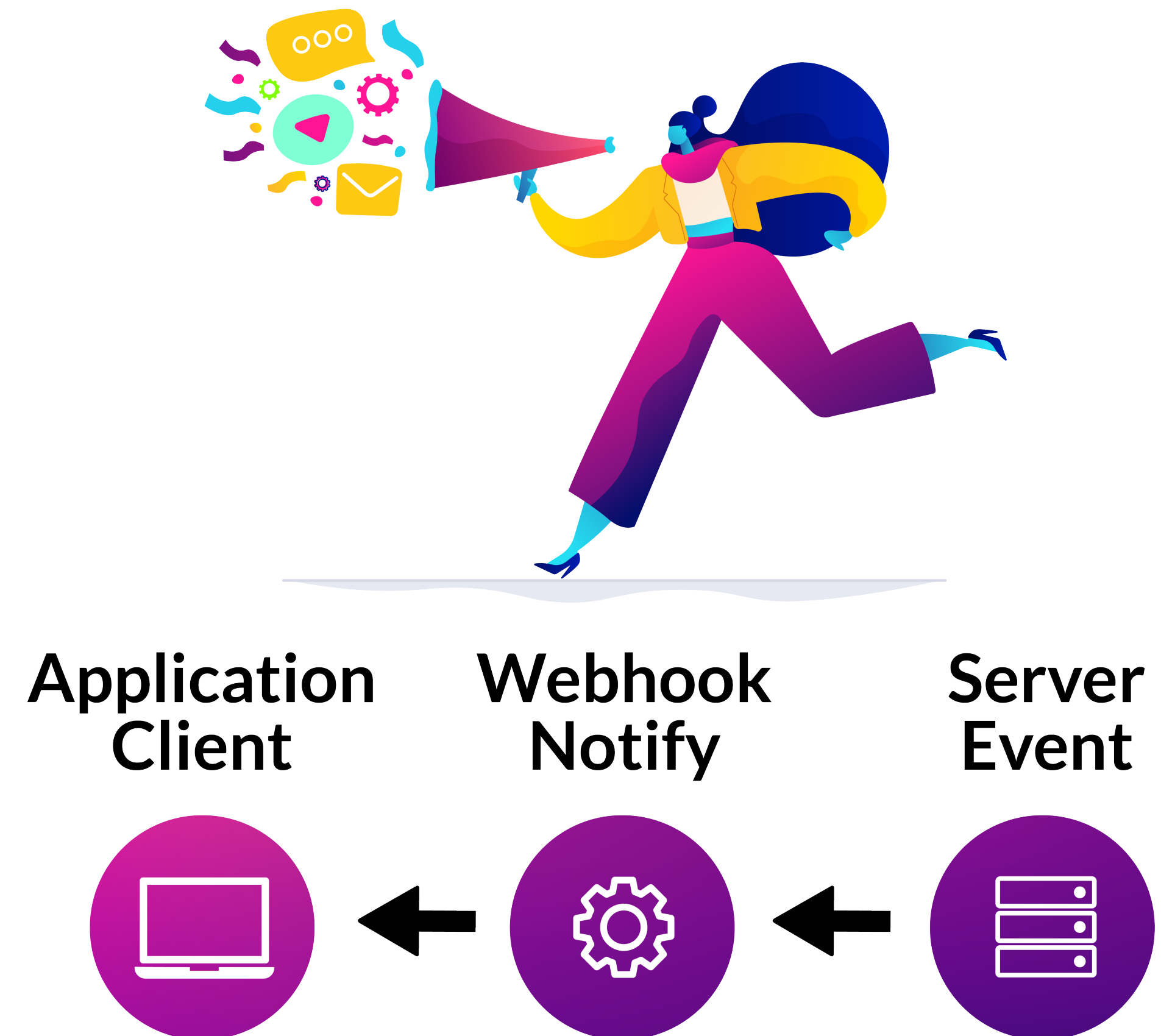


- Notifications (let somebody or something know when something happens)
- Other dependent workflows (have to happen in order, across multiple apps)
- Synchronized data (two or more apps share the same data, and it can be edited)

How are Webhooks Different?

Webhooks are like a reverse API

- Removes need for client to recognize events using polling
- Server sends data to client when events happen
- Clients can subscribe only to the events they care about
- Workflows advance right away
- Webhooks can usually be found in the notifications settings



Apps that offer APIs

Some have webhooks also, many more





No code integrations

No Code Integrations

Make use of APIs and Webhooks without coding




- make apps talk to each other, even when they aren't on the integrations list
- introduce some of your own conditions to define how data is passed on
- reformat the data from one app so that the receiving app can use it

- create complex workflows that involve more than 2 apps, or loop back between apps
- pass data from an app to itself to create new internal workflows
- there are other options similar to zapier and IFTTT

Example of Zapier Integration

Urgent ticket posts to Slack channel




 **Millenium Falcon out of gas**


Today at 09:27am

- han.solo@theresistance.com
- Via help@zendesk.com

[\(change\)](#)

**Public reply**Internal note

Leia is stuck with the Ewoks on Endor. I need to get there in under a parsec!

 **URGENT SUPPORT TICKET 3962** APP 9:27 AM

Han Solo

<https://ripeda.zendesk.com/agent/#/tickets/3962>

Millenium Falcon out of gas

Example of Zapier Integration

Urgent ticket posts to Slack channel



What do you want to automate today?

When This Happens ...

Search & Select ...

... Do This!

Search & Select ...

MAKE THIS ZAP

Example of Zapier Integration

Urgent ticket posts to Slack channel



What do you want to automate today?

When This Happens ...

Search & Select ...

▼

... Do This!

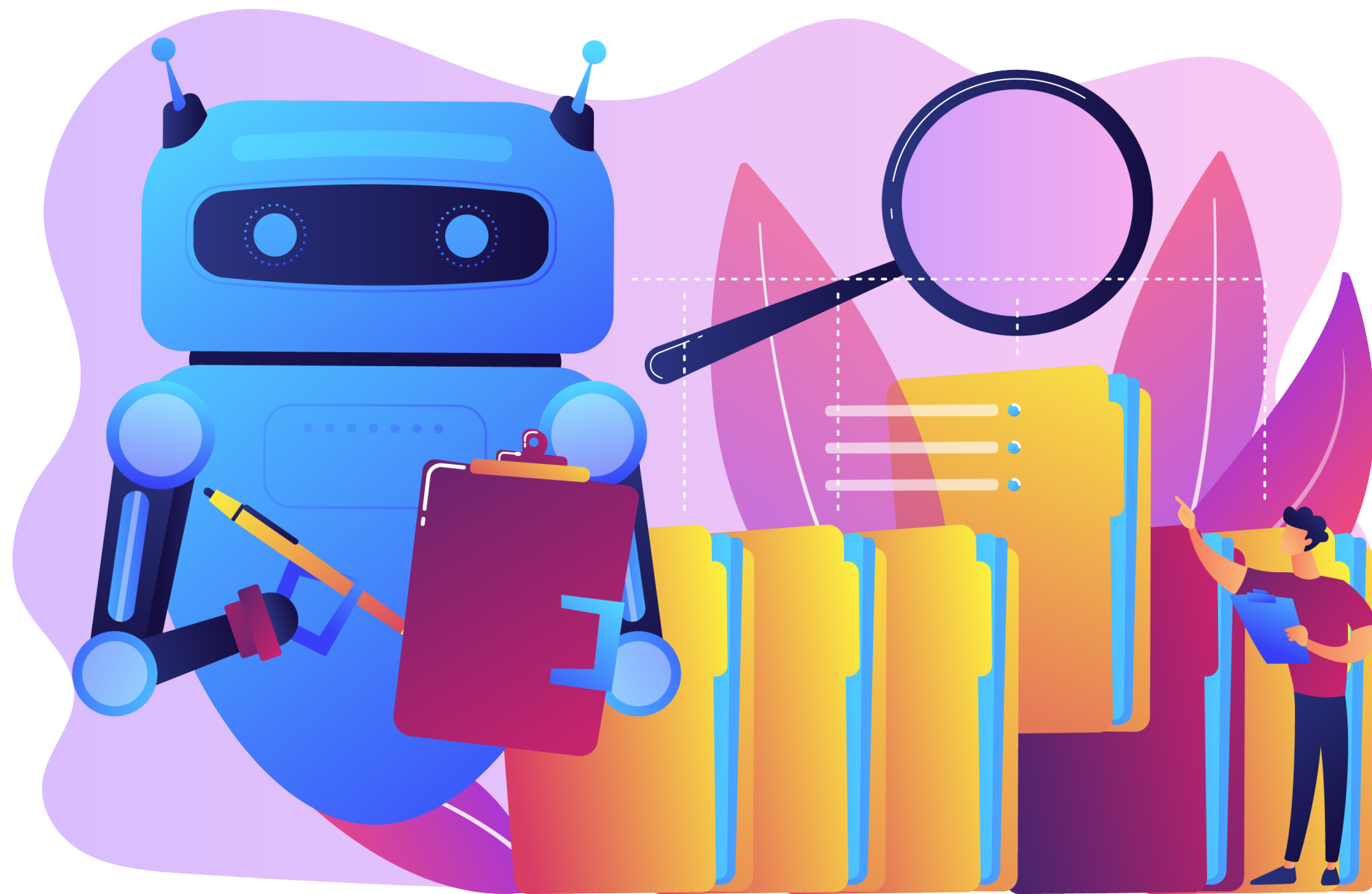
Search & Select ...

▼

MAKE THIS ZAP

When What You Need Doesn't Exist

Custom integrations are a bigger investment



Things to consider...

- Improve accuracy, reduce human error?
- Improve experience for end users or staff?
- Save time? How much?
- Cost/effort to maintain?

500 manual tasks/month x 8 minutes/task = 70 hours/month

6 errors/month x 30 minutes/error = 3 hours/month



Using REST APIs

Anatomy of REST API Requests

Read, write and update data through structured URLs



GET

Retrieve



POST

Create



PUT

Overwrite



PATCH

Update



DELETE

Remove

`http://api.example.com/v1/users/{user-id}/accounts`

Get devices - `http://api.example.com/v1/managed-devices`

Get device detail - `http://api.example.com/v1/managed-devices/{device-id}`

Get users - `http://api.example.com/v1/users`

Get user detail - `http://api.example.com/v1/users/{user-id}`

Intro to JSON Objects

A popular format for requests and responses

```
let obj = {  
  username: "xyz",  
  password: "xyz@123",  
  email: "xyz@xyz.com",  
  uid: 1100  
}  
  
console.log(obj.username)  
> xyz  
console.log(obj.uid)  
> 1100  
obj.uid = 2200  
console.log(obj.uid)  
> 2200
```

```
let listObj = [  
  {  
    username: "xyz",  
    uid: 1100  
  },  
  {  
    username: "abc",  
    uid: 2200  
  }  
]  
  
console.log(listObj[0].username)  
> xyz  
console.log(listObj[1].username)  
> abc
```

Tinkering with REST APIs

REST requests are sent through a set of URLs

GET

<http://api.tvmaze.com/search/shows?q=the%20%good%20%place> → "id": 2790

GET

<http://api.tvmaze.com/shows/2790/episodes> → "name": "Flying",
"airdate": "2016-09-19",
"summary": "Eleanor tries to prove to ..."

GET

<http://api.tvmaze.com/shows/2790/cast> → "id": 27257,
"name": "Kristen Bell"

GET

<http://api.tvmaze.com/people/27257/castcredits> → "show": <http://api.tvmaze.com/shows/134>
"character": "<http://api.tvmaze.com/characters/36509>"

GET

https://api.nasa.gov/mars-photos/api/v1/rovers/curiosity/photos?sol=1000&page=2&api_key=DEMO_KEY



Configuring Webhooks

Configuring a Webhook Target

Some are trivial, some apps have more flexibility

All require a destination (aka callback or target) URL

The screenshot shows a 'Trigger Webhook' configuration panel. It includes a 'Request Type' dropdown set to 'POST', a 'Callback URL' field with the value 'https://acme.freshdesk.com/api/v2/tickets/{{ticket.id}}/notes', and a checkbox for 'Requires Authentication'. Below this is a 'Custom Headers' section with a text area containing 'X-Sample-Header1: ABCDE'. The 'Encoding' section has three radio buttons: 'JSON' (selected and circled in purple), 'XML', and 'X-FORM-URLENCODED'. There are also 'Simple' and 'Advanced' radio buttons, with 'Advanced' selected. The 'Content' section contains a text area with a JSON payload:

```
{  "body": "Hi Tom, has your issue been fixed ?",  "private": false,  "notify_emails": ["tom@yourcompany.com"]}  
```

. Annotations with purple arrows point from external text to the 'Callback URL', 'Custom Headers', 'JSON' encoding option, and the 'Content' text area.

Trigger Webhook

Request Type: POST

Callback URL * [How to use Webhooks >](#)

[Insert Placeholder](#)

☐ Requires Authentication [Remove Custom Headers](#)

Custom Headers

X-Sample-Header1: ABCDE

Enter one custom header value pair per line.

Encoding

☒ JSON ☐ XML ☐ X-FORM-URLENCODED

☐ Simple ☒ Advanced

Content

```
{  "body": "Hi Tom, has your issue been fixed ?",  "private": false,  "notify_emails": ["tom@yourcompany.com"]}  
```

[Insert Placeholder](#)

This section lets you write custom API requests. Click on the Insert Placeholders button to include ticket details such as Requester Name, Priority, Ticket Status etc. with your request.

[Webhook Usecases](#)

Some allow custom headers

Most allow you to specify the format

Few support custom payload
(but this may eliminate need
for middleman app)

Exploring webhook payloads

Does it have what you need? Find out first

https://webhook.site

Webhook.site allows you to easily test webhooks and other types of HTTP requests. [What is a webhook?](#)

Any requests sent to that URL are logged here instantly — you don't even have to refresh!

Here's your unique URL that was created just now:

`https://webhook.site/8f437687-7827-4c14-893e-878615b4e987`

 [Copy to clipboard](#)  [Open in new tab](#)

If you want to edit the response (status code, body content), click Edit above.

Bookmark this page to go back to the requests at any time.

Click **New URL** to create a new url with the ability to customize status code, response body, etc.

[Star on GitHub](#)



Build a custom endpoint



What You Need to Build a Live Endpoint

Custom endpoints can be simplified with AWS Lambda

Requirements:

1. A public URL to be the endpoint
2. A service listening on HTTP ports
3. Software to process the payload and take action

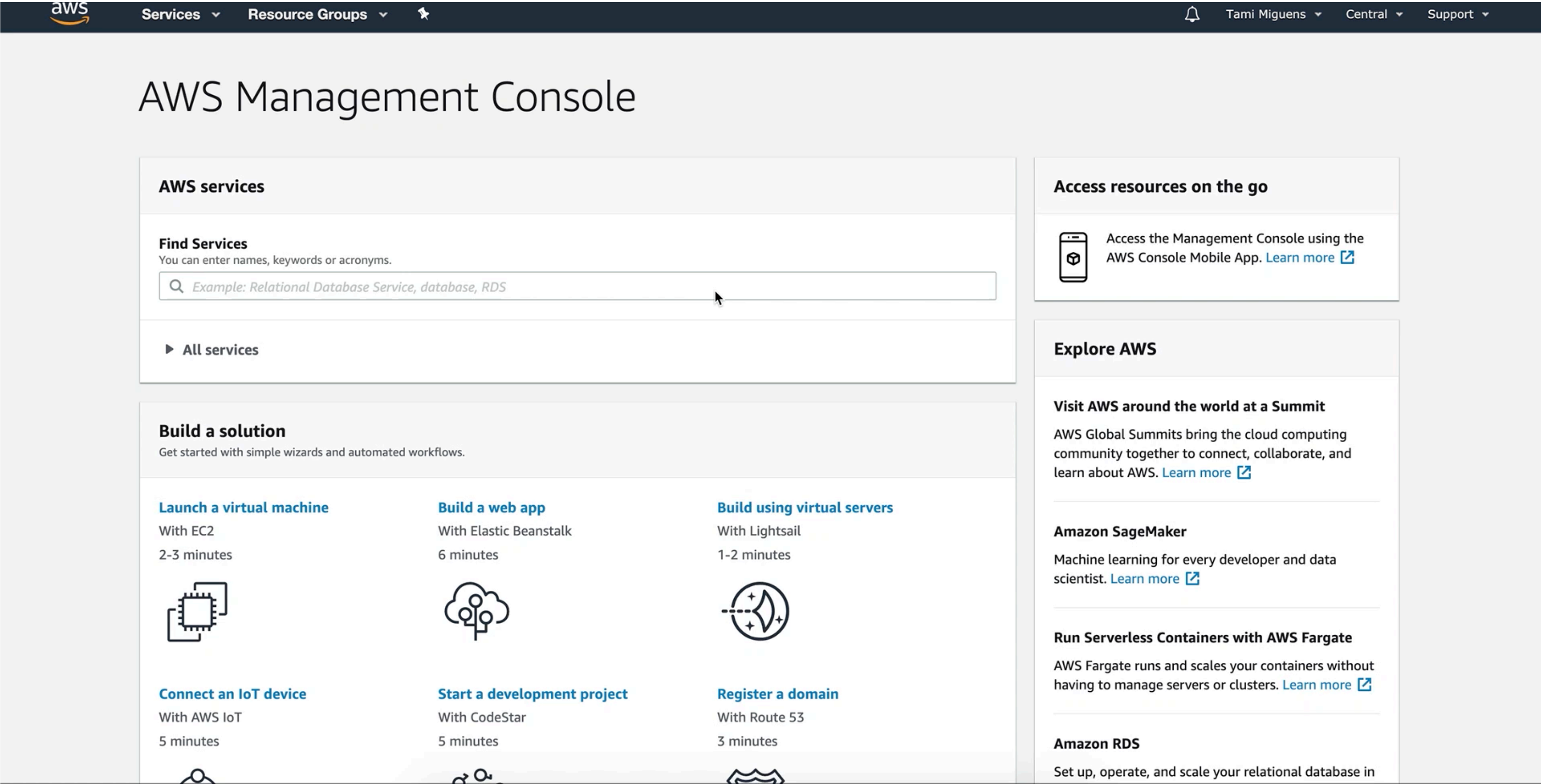
<https://s3.amazonaws.com/lambda-tools/pricing-calculator.html>

<https://aws.amazon.com/getting-started/tutorials/run-serverless-code>



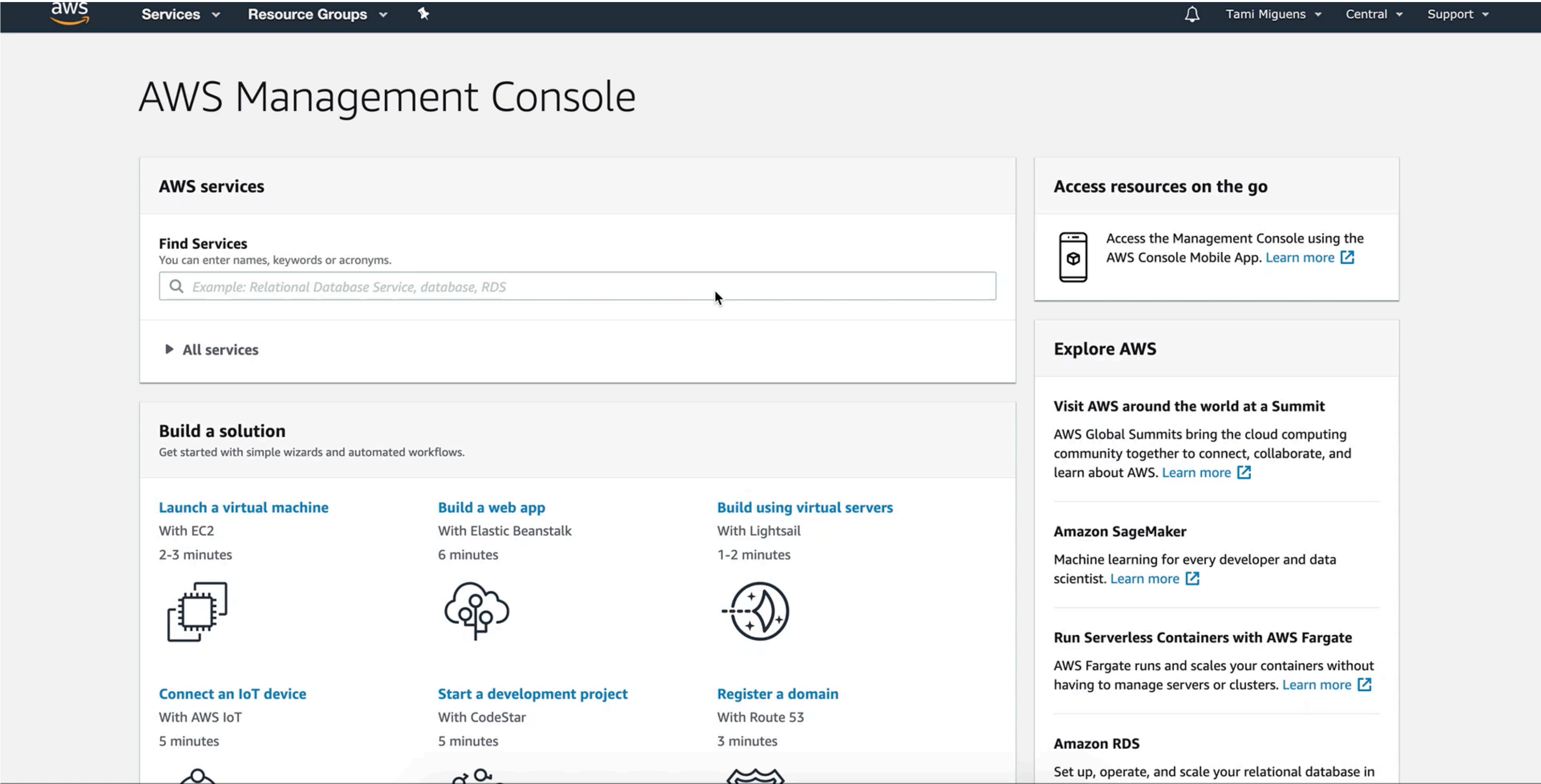
What You Need to Build a Live Endpoint

Custom endpoints can be simplified with AWS Lambda



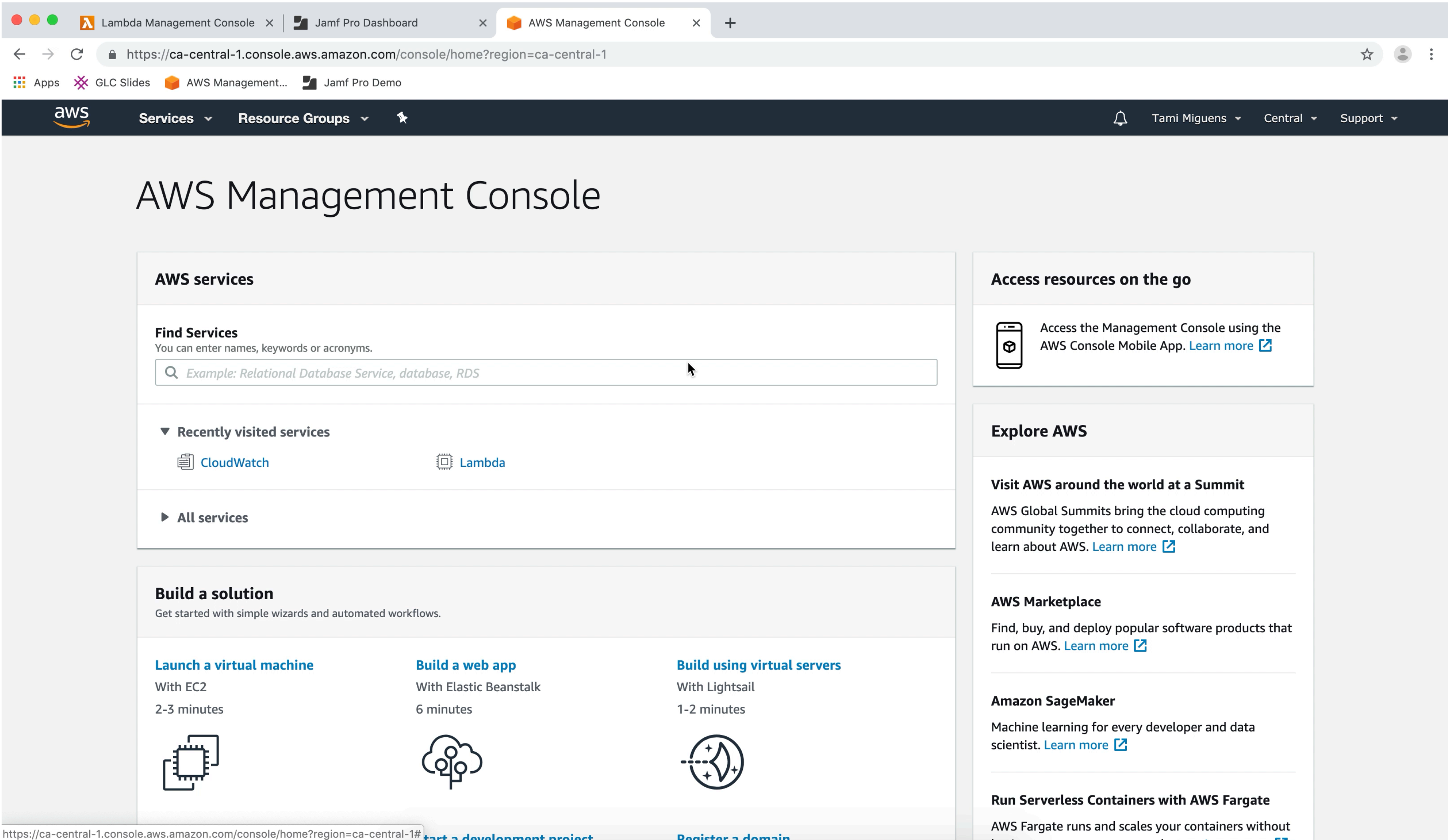
What You Need to Build a Live Endpoint

Custom endpoints can be simplified with AWS Lambda



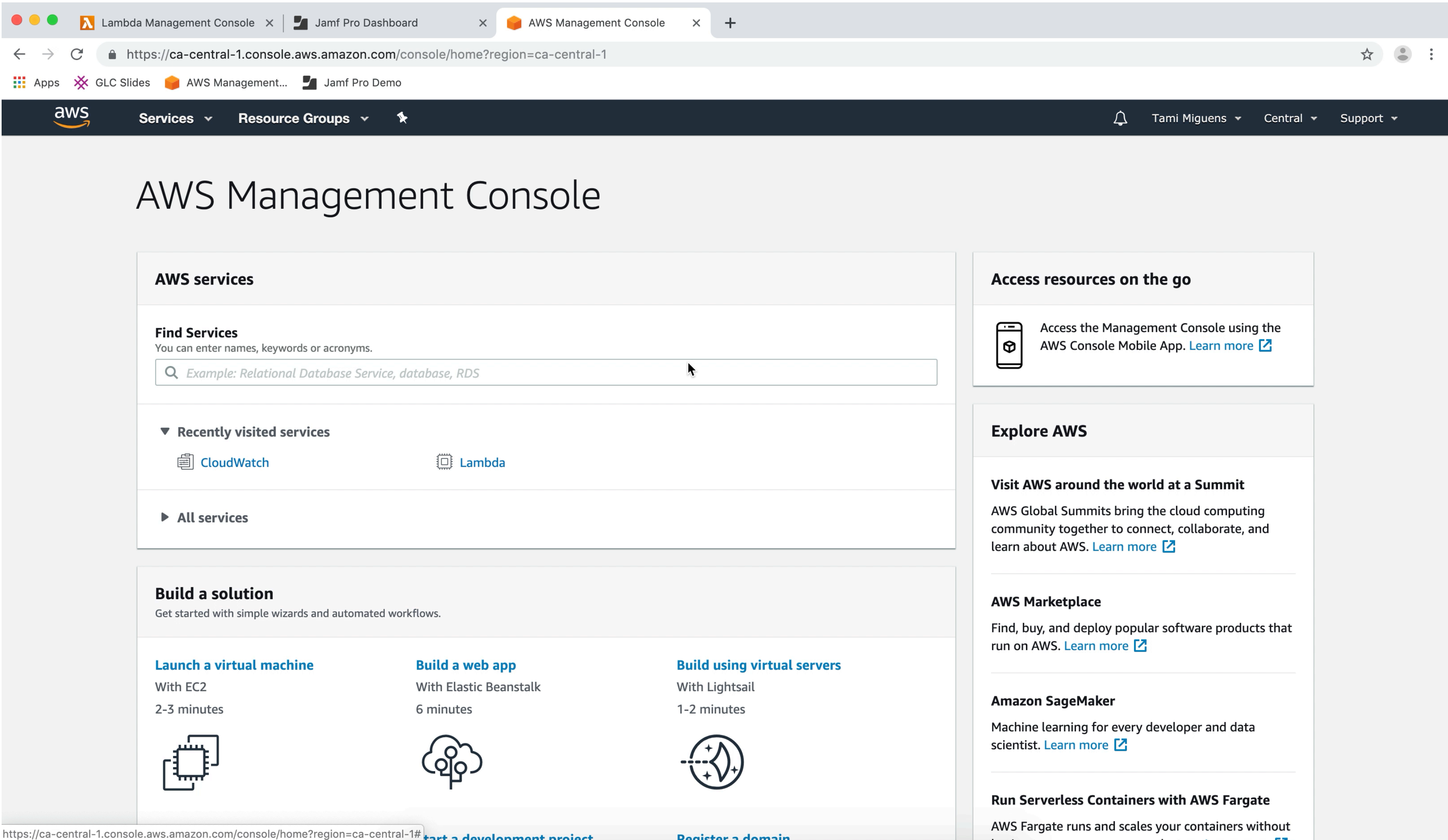
What You Need to Build a Live Endpoint

Custom endpoints can be simplified with AWS Lambda



What You Need to Build a Live Endpoint

Custom endpoints can be simplified with AWS Lambda



Get in Touch



Tami Miguens



RIPEDA Consulting is a dedicated team of experienced engineers and IT professionals in Calgary, Alberta. We offer managed services and IT consulting for Mac and Windows computers and servers, as well as secure, high-performance networks for businesses, institutions and schools.

