IT/Dev CONNECTIONS

GOING FULLY SERVERLESS

Is it possible to never worry about servers?

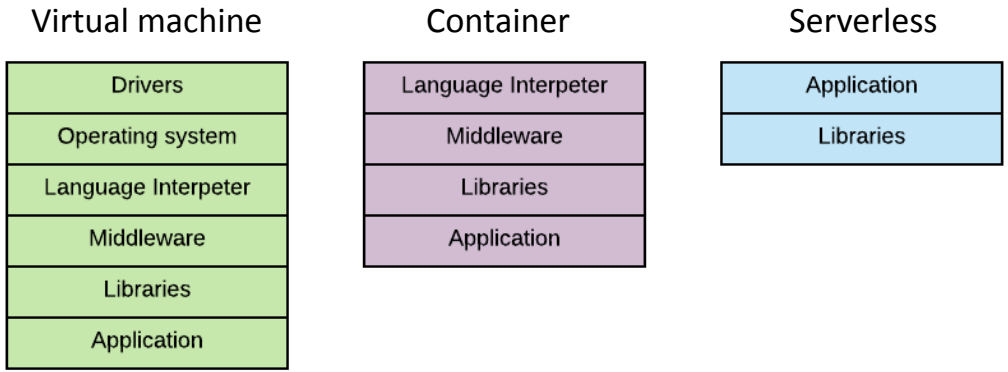
Jonathon Valentine
CTO
ThingCo

Delivered by KNect365 TMT an Informa business

Who am I?

- CTO & Co-founder of ThingCo, using next generation telematics and IoT to use driving data to help consumers understand the value of their data, while making them safer drivers, and helping them if they have an accident
- First employee at insurethebox, a ”Black box” car insurance company that used telematics in all end to end operations, selling over 800,000 insurance policies, collecting 4 billion miles of driving data, and insuring 60% of all 17 year old drivers in the UK.
- Designed and built the world’s first, award winning, telematics counter fraud platform

I've got some code, where to run it?



Why though?

- Auto scaling
- No security patches or OS updates
- No load balancing to worry about
- Guaranteed environment
- No container building
- Cheap!* if done right!

Serverless isn't "server less"

- Creating a serverless architecture doesn't remove servers, it just removes the responsibility of them, pushing that task to AWS, Azure or GCP instead
- Servers still run your code, but as the resource pool is much greater then your own on-premise or virtual machine pool, scalability no longer becomes an issue
- Going fully serverless means not relying on a Kubernetes cluster in a virtual machine pool, the scaling of your code
- Chef's don't want to raise their own cows to make great steaks, leave that to the farmers!

Serverless doesn't have to be code

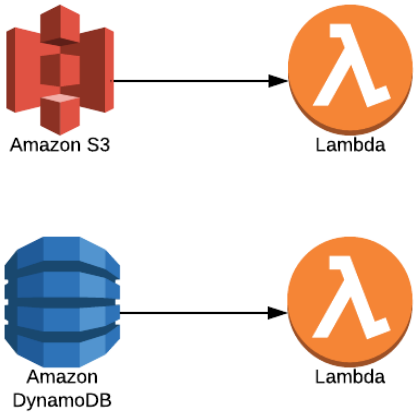
- Lambda – running code
- DynamoDB / Aurora – database as a service
- IoT – MQTT device interface as a service
- SQS – Message queues
- CodePipeline / CodeBuild – Continuous integration as a service

All these are examples of things a company would usually manage themselves

Things to consider

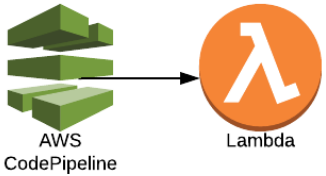
- Serverless doesn't mean you can scale to infinity – it just moves the bottleneck, AWS Lambda may be handle 1 million requests of your function at once, but can the database that function accesses?
- Developers don't usually write with speed at the forefront (I know of a developer who wrote a job that would run for 36 hours every 24 hours…)
- AWS charges in 100ms segments, Lambda is cheap, but not that cheap with bad code
- Serverless doesn't remove security risks, it just takes advantage of your provider's security, still think about policies, api keys etc.

Where can you use serverless?



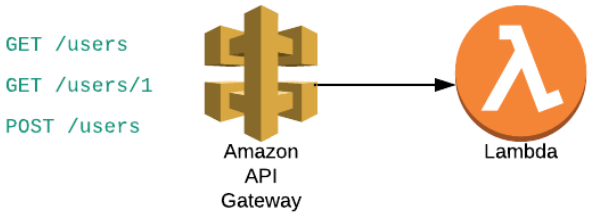
- Functions can be called on event such as a file been added to a bucket
- This also works for something changing in a database
- Given an external party access to push files to a bucket? Use the object created event to run the function, rather then having a scheduler keep checking and costing money / resource

Where can you use serverless?



- You can also run functions as part of your deployment processes
- For example, at ThingCo, we've got a lambda function that's sole job is to approve or decline a deployment based on the environment and other factors

Where can you use serverless?



- Lambda functions can also replace your back end web services
- By using an API gateway to handle requests, the lambda can talk to your databases and return the output in what ever format you need
- Takes advantage of the security layers in API Gateway
- Not paying for a server 24/7 if requests only come in during office hours!

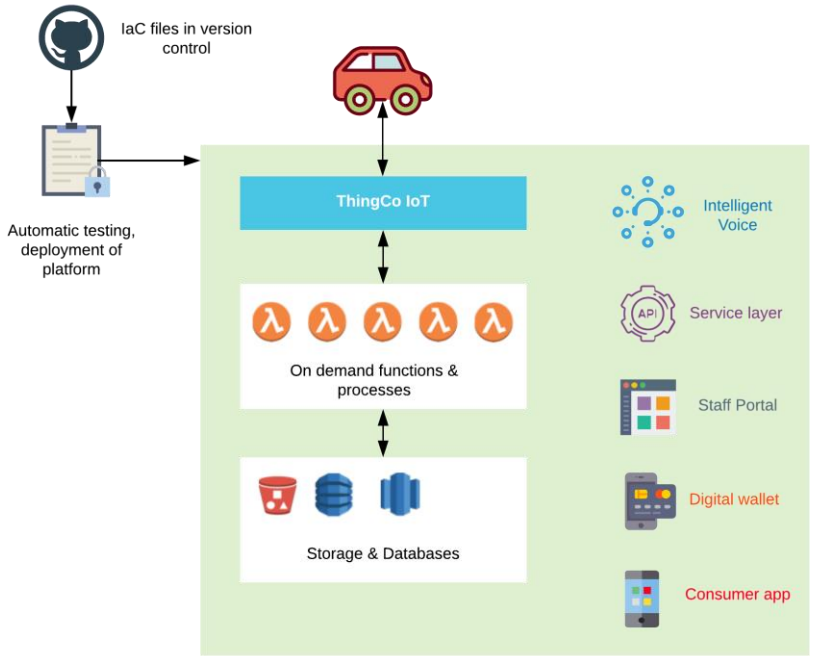
Is the price advantage worth the move?

- $0.20 per million requests
- $0.00001667 per gb-second of compute (rounded to nearest 100ms)

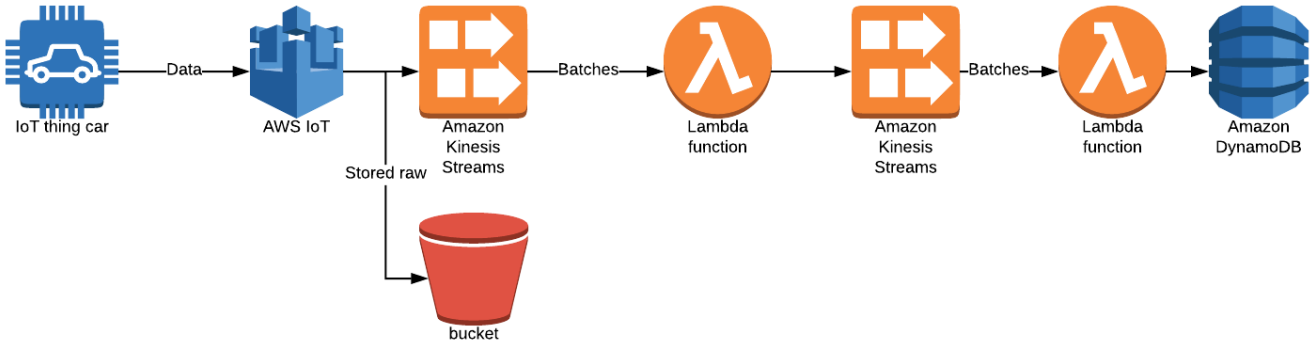
Trek10 did some great work comparing virtual machine versus lambda (https://www.trek10.com/blog/lambda-cost/)

- A lightweight backend with 10,000 requests a day would cost $0.31 a month, a 10^{th} of the cost of the cheapest virtual machine
- A scheduled job running using 1gb of memory every hour for 2 minutes would cost $1.44 a month, 1/3 of the cost of a t2.nano which only has 0.5gb of memory
- Even a big workload, using a m4.large, would require 18 requests a second to be cheaper

So how does ThingCo go serverless?



So how does ThingCo go serverless?



ThingCo's full end to end solution is priced on a per car model averaging $0.02 per car per month

Let's build something

I've done talking now, I want to show you how it's easy to build something that is stable, scalable and cheap!

Thankyou!

@jvthing