

# Serverless: A Sane Approach to Infrequent Tasks

...

Or a path to dropping your AWS bill by a zero

By Jason Price  
@japrice

# What is Serverless?

It is a terrible name.

Get over it, because we're stuck with it.

# High level description?

Run your code in response to some event.

This is the key to doing serverless well

# What's an event? Nearly anything.

- Upload a file to s3
- Add item to SQS queue
- Some other Lambda executing
- Kinesis event
- Incoming web request
- Clock strikes 12
- Cloudwatch metric goes over some value
- New VPC created
- New user registered in Cognito
- Device registers in IoT
- Gnat farts in the datacenter

Etc, etc, etc.

# What does Marketing call it?

What do the cloud vendors call it?

- AWS: Lambda
- GCP: Google Cloud Functions
- Azure: Functions
- Kubernetes: kubeless or fission

# Example Use Case:

I'm a nurse with a dumb tablet. I update a chart, and upload the new chart to S3.

Lambda is triggered by the S3 upload to process the chart. This begins the web of business logic associated with a chart update:

- 1) Send to Primary Care Doctor for review
- 2) Dose of medicine administered? Update billing, and inform insurance co.
- 3) Were they released? Close out the bill, update insurance co, send bill to customer
- 4) Scan for missing info. Field missing? Update chart with question to be answered
- 5) etc

# Structure of a Lambda

1. You make a zip file, with your python code in it.
2. You nominate one file, and a function in that file as the “lambda\_handler”
3. The function takes two arguments: “event” and “context”
4. Event is the dictionary for the JSON fed into the function.
5. JSON is arbitrary
6. Language options:
  - a. .Net v1, v2, v2.1
  - b. Go v1.x
  - c. Java v8
  - d. Node.js v4.3, v6.10, v8.10
  - e. Python v2.7, v3.6

# Example: Hello World

Live Demo

Code here:

[https://github.com/zapman449/pyatl-lambda-api-gateway/tree/master/01\\_hello\\_world](https://github.com/zapman449/pyatl-lambda-api-gateway/tree/master/01_hello_world)



# Key Takeaways of hello\_world

- 1) Making a lambda is trivial
- 2) Need to have a pre-built Lambda Role which can do CloudWatch Events and Logs
- 3) The cloud9 IDE is pretty decent. Sane editor, built in harness for running tests, log viewer for test runs built in...
- 4) To play in this world, you'll need a moderate understanding of AWS IAM as well. Be aware.

# Example: Increment a Thing

Live Demo

[https://github.com/zapman449/pyatl-lambda-api-gateway/tree/master/02\\_increment\\_counter](https://github.com/zapman449/pyatl-lambda-api-gateway/tree/master/02_increment_counter)

# Key Takeaways

- 1) I've got an API running, and there's not a server in sight. No root logins, no configuration management, no ops team, no SSL certs...
- 2) Conversely, I don't have a high functioning metrics, monitoring or logging world setup either
  - a) Note: there are companies rushing to fill this void. If you're looking, I suggest IOPipe, but they have competitors
  - b) I mean, you can GET the logs... but it's not easy to say "give me all WARNINGS from 09:12-09:15 this morning"
- 3) You can wire up a highly sophisticated web API with very little effort
- 4) Or, you can wire up the lambda to a LOT of services within AWS with little effort

# Lambda Limitations

Nothing's perfect:

- Max runtime is 5 minutes (default is 3 seconds)
- Max RAM usage is 3gb (default is 128mb)
- Billed by the microsecond \* ram usage
- Max zip file size is 50mb
- There are limits to # of concurrent jobs. By default it's 1000.
  - on the plus side, there's a neat way of saying "only instance of this can run": set concurrency to 1
- "Cold Start Problem"

NOTE: I expect this slide to be out of date by the end of 2018

# Lambda Limitations

## “Cold Start Problem”

If your job hasn't run for a while, Lambda needs to pull your zip file from S3 and then launch the handler for the job. This can add latency to critical, but infrequent operations.

# Use Case: ECR Registries

We have tiny lambdas to create registries, create namespaces, list all images in a repo, etc.

We made a docker image which can use your IAM rights to call the functions.

Keeps name sprawl under control, gives teams easy access to what they need.

We've got a "reconcile\_policies" which does what you expect. Started taking >2 minutes to run... so we rewrote it to be multi-threaded, and it now runs in 18 seconds.

# Bold Assertions

“Sane approach to infrequent tasks”

Grade: A+ -> Pretty solid footing.

NOTES: Easy to do something Cron like. Also straightforward to do something more complicated around job dependencies and event chains.

# Bold Assertions

“AWS is ahead of everyone else by at least 2 years”

Grade: B -> Ill defended, but manifestly obvious

NOTES: AWS launched the whole “serverless” movement with Lambda. No one else had ever done such a thing at scale.

Don't be confused by the fact that Google, Azure, Kubernetes, etc have Serverless frameworks. They are nice, but the KEY to serverless is the portfolio of event triggers. Kubernetes events are “arbitrary HTTP requests”, which is nice... but not as rich with options as AWS with even S3 uploads or SQS queue publish



# Bold Assertions

“Key to serverless done well is keying off events”

Grade: B- -> Barely mentioned, and only hinted at.

NOTES: Ultimately, this gets deep into your architectural toolbox... and you need to know your needs and how this might or might not help you.

But if you're leveraging major pieces of AWS like SQS, SNS, kinesis, etc... this needs to be in your toolbox

# Bold Assertions

“Drop your AWS bill by a zero”

Grade: C+ -> barely sufficient effort, needs more polish

NOTES: Lambda is a great way to radically reduce your EC2 bill. However, it's NOT a cure-all. Your RDS / Data-Storage bill will probably remain the same, and honestly, most anything not EC2 will be unaffected. And your API Gateway bill will probably go up. It's billing model is... complex

It seems like a lot of companies AWS spend is about 40% RDS, 50% EC2 and 10% OTHER, +/- 10% for each. So going “all in” on Lambda could drop your bill by 30ish%... which is nothing to sneeze at, but not “drop your bill by a zero”.

# Final Slide:

Code can be found here:

<https://github.com/zapman449/pyatl-lambda-api-gateway>

Any questions?