

# Provider-Side Serverless Opportunities

Garrett McGrath



Serverless Opportunities



- Serverless computing is the first primitive where the provider understands the applications it runs
  - Relinquishes control of application provisioning and scaling to cloud provider
  - Forces applications to explicitly declare events
  - Higher level abstractions declare interactions between Serverless components (AWS StepFunctions, Microsoft LogicApps)

#### **Behavior Prediction**

- When a function will be executed
- When data will be fetched (IAM checks, database requests, ect.)
- Automatic CPU/memory allocations
  - Azure Functions already bills for memory used





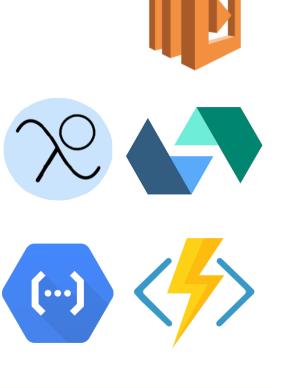


## Economic Pressures

- Cloud providers need to cost-compete by running datacenters more efficiently (utilization, energy-efficiency)
- Cloud customers seek to reduce cost by minimizing resource waste
  - Want needs of application to match allocated resources
  - Serverless computing is a large step forward

#### We're Not There Yet

- "Never pay for idle, or for wait" Ben Kehoe (Serverlessconf '17)
  - Time spent waiting on network (function executions or otherwise) is wasted by both provider and customer
  - Provider must be allowed to free function resources in interim
    - What state to persist? (New APIs?)





### **Serverless Opportunities**



#### **Serverless State and Communication**

- "Permanent storage lives elsewhere" Serverless Manifesto
  - But stateful and/or communicative scenarios exist beyond simple fan-outs
    - How to enable stateful services in serverless platform?
    - Inter-function communication and addressing?
    - Networking-sensitive function placement (HPC scenarios)
  - Not permanent, but temporarily persistent and discoverable

#### **Takeaways**

- Serverless computing allows providers to understand customer applications, and to deliver value based on this information
  - Presents many design challenges
- Many scenarios/paradigms remain unexplored by providers



