THE UNIVERSITY OF CHICAGO

MPCS 51033 • AUTUMN 2017 • SESSION 10

BACKENDS FOR MOBILE APPLICATIONS

CLASS NEWS

CLASS NEWS

- No office hours Thursday
  - Email, slack, etc.
- Final Projects due December 7th at 11:59 pm.

SERVERLESS ARCHITECTURES

SERVERLESS ARCHITECTURES

- Serverless architectures can mean many things
  - BaaS
  - PaaS
- Function as a service (Faas) is the extreme realization of this architecture

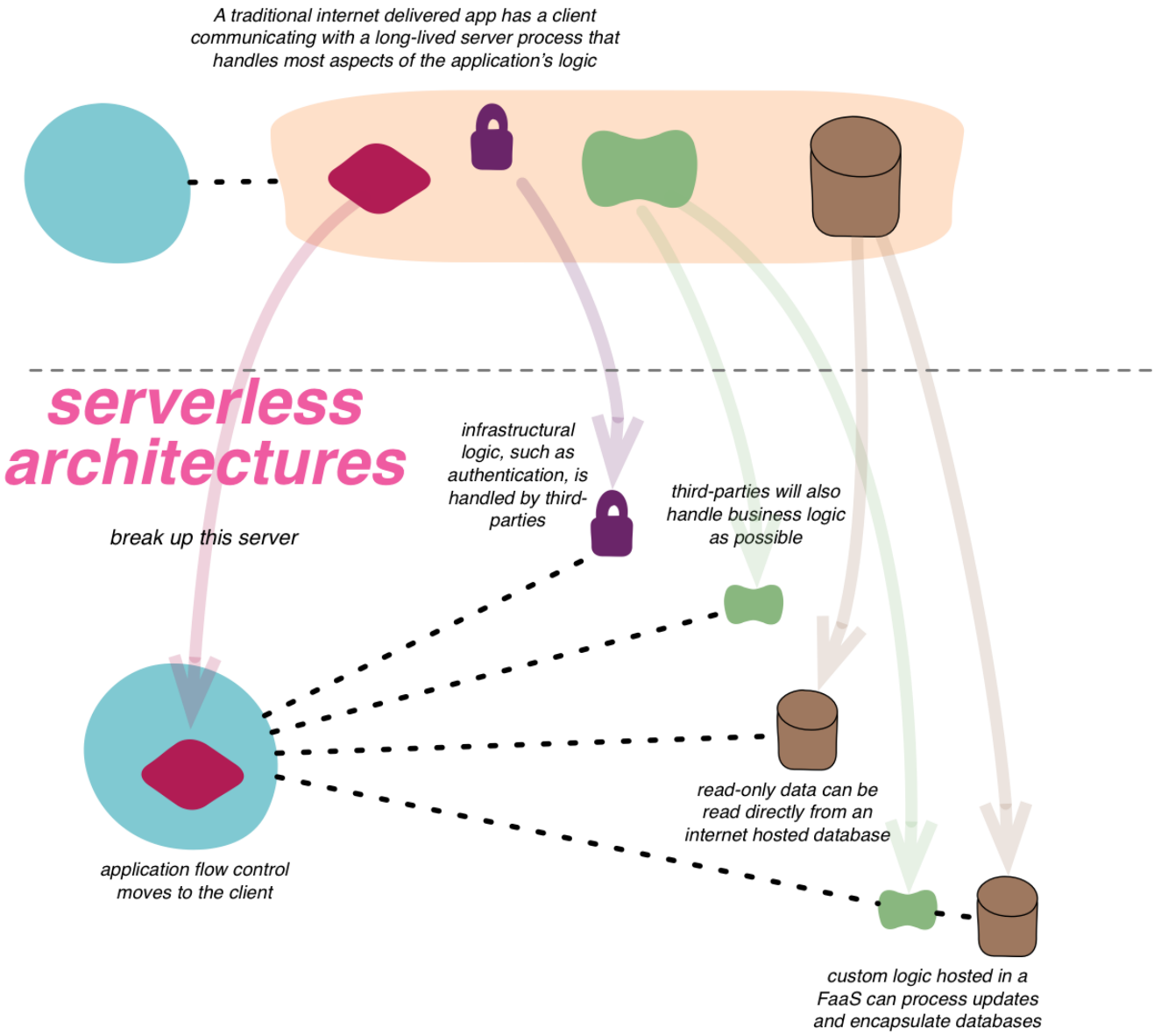
I don't think anyone really calls it this

SERVERLESS ARCHITECTURES

- Despite what you may think, serverless architecture is really about moving as much behavior to the front end as possible
- Remove the need for traditional "always on" server

SERVERLESS ARCHITECTURES

- Traditional vs. serverless architectures

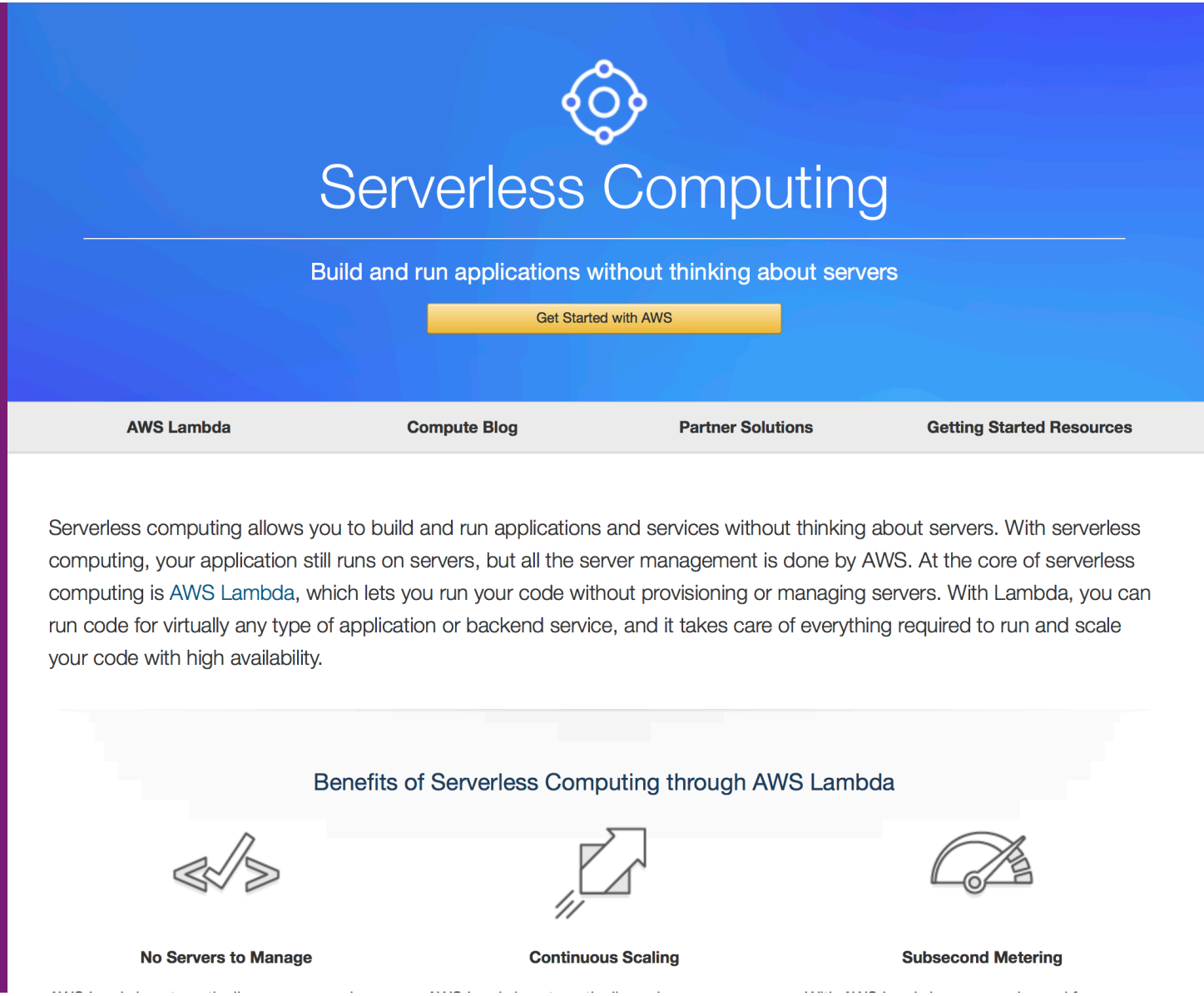


SERVERLESS ARCHITECTURES

- Serverless applications have to rely on third-party services to accomplish tasks that are typically consolidated by a server
  - Each service might be provided by a different provider (eg. authentication with Google, database with Azure) or by a single provider (eg. Firebase)
  - The fact that some services are by the same provider is more of an administrative convenience...serverless is about abstraction

SERVERLESS ARCHITECTURES

- Key players
  - AWS Lambda
  - Microsoft
  - Google Cloud Functions
  - Firebase Cloud Functions



SERVERLESS ARCHITECTURES

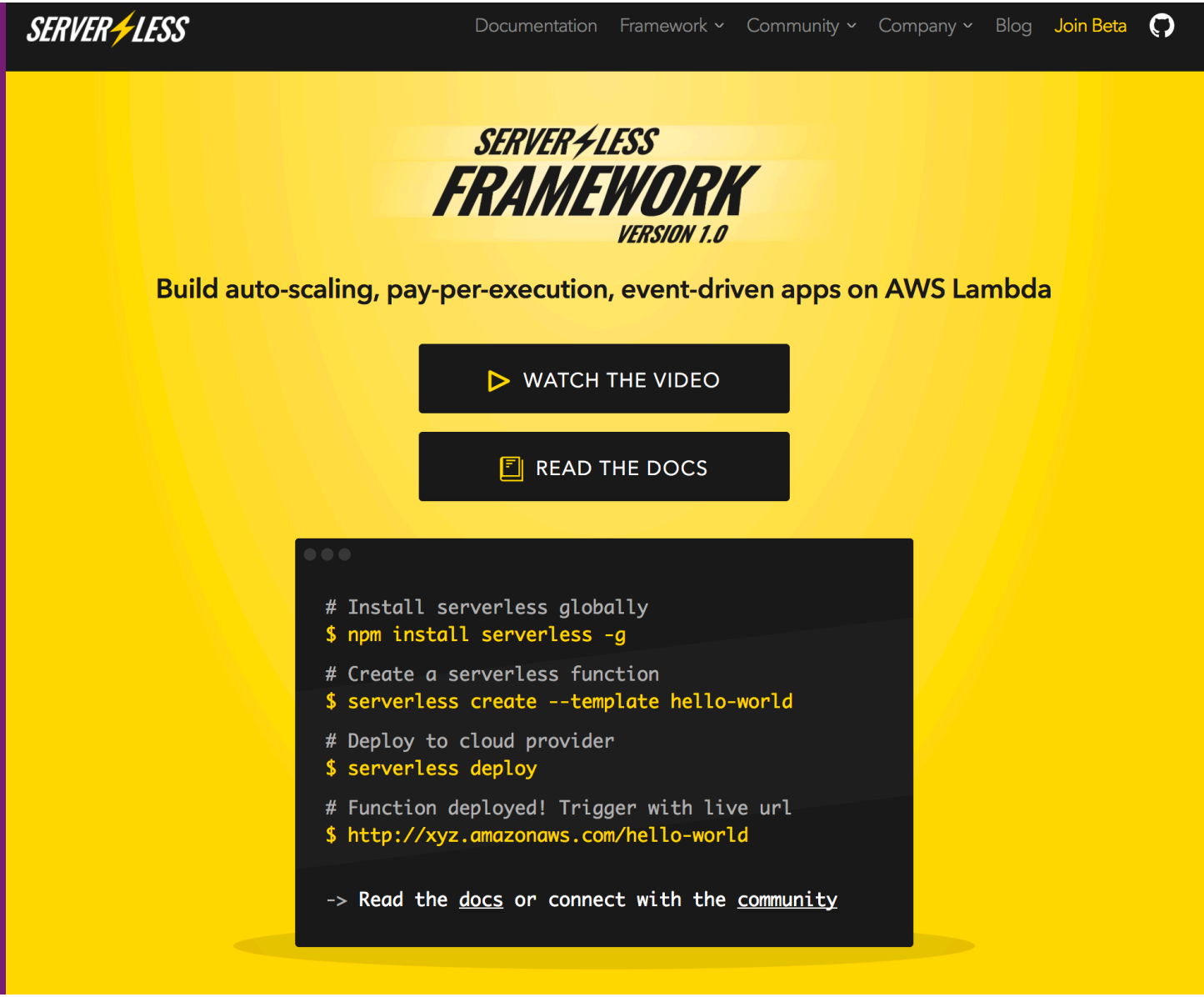
- Serverless logic tied to services
  - Parse (RIP)
  - Realm
  - Firebase

Go to News | Transcript | About the Speaker

Today we're announcing Realm Functions, a new part of Realm that makes building server-side functionality a lot easier for mobile developers. Now, you can make server-side features without enlisting backend developers, plus you get all the benefits of building on top of the Realm Mobile Platform: you don't need to add another endpoint to a server, and then write the serialization and networking code that would let you connect with it. You just connect your app to Realm, write a Realm Function in your web dashboard, and watch your code execute reactively as data streams in. Today's release is a beta, and it's available today to everyone, whether you're building an app in an enterprise-scale team or for a small side project.

SERVERLESS ARCHITECTURES

- Frameworks for server less
  - Write once, deploy everywhere



SERVERLESS ARCHITECTURES

- A major consideration
  - The runtime of server less functions can be extremely limited
  - Server only active during process (pay per time)
  - Data can be returned in the response
  - Data saved requires an outgoing request

All about tradeoffs

SERVERLESS ARCHITECTURES

- Benefits
  - Costs (pay for what you use)
  - Reduced complexity (of running a server)
  - Scaling

SERVERLESS ARCHITECTURES

- Costs
  - Conceptual
  - Third-party dependency
  - Control (whims of big companies)
  - Local development and testing

AWS LAMBDA

AWS LAMBDA

Compute

EC2
EC2 Container Service
Lightsail
Elastic Beanstalk
Lambda
Batch

Developer Tools

CodeStar
CodeCommit
CodeBuild
CodeDeploy
CodePipeline
X-Ray

Analytics

Athena
EMR
CloudSearch
Elasticsearch Service
Kinesis
Data Pipeline
QuickSight

Application Services

Step Functions
SWF
API Gateway
Elastic Transcoder

Storage

S3
EFS
Glacier
Storage Gateway

Management Tools

CloudWatch
CloudFormation
CloudTrail
Config
OpsWorks
Service Catalog
Trusted Advisor
Managed Services

Artificial Intelligence

Lex
Polly
Rekognition
Machine Learning

Messaging

Simple Queue Service
Simple Notification Service
SES

AWS has a lot of offerings

Database

RDS
DynamoDB
ElastiCache

Internet Of Things

AWS IoT

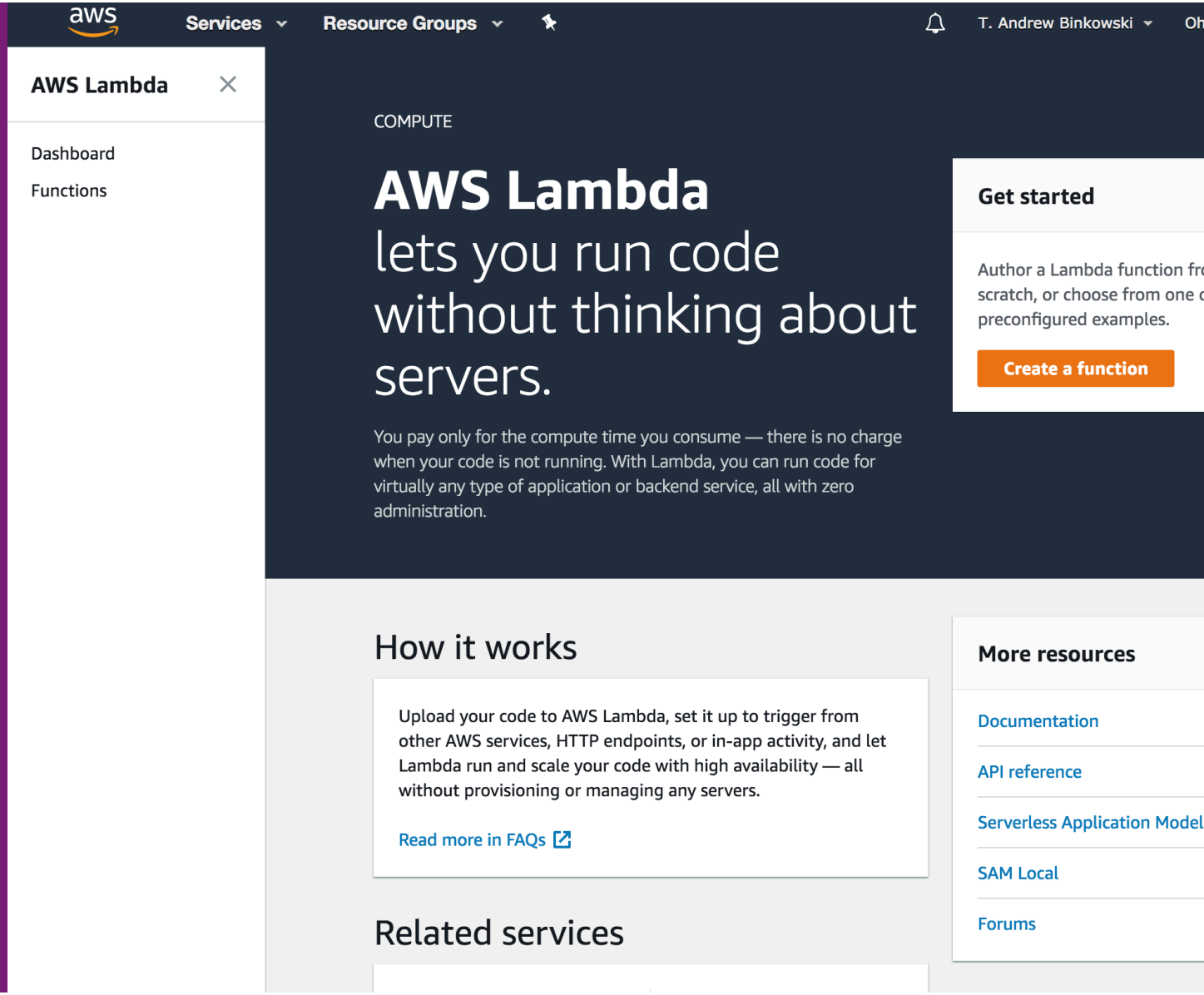
Business Productivity

WorkDocs
WorkMail
Amazon Chime

Desktop & App Streaming

AWS LAMBDA

- AWS Lambda serverless architecture



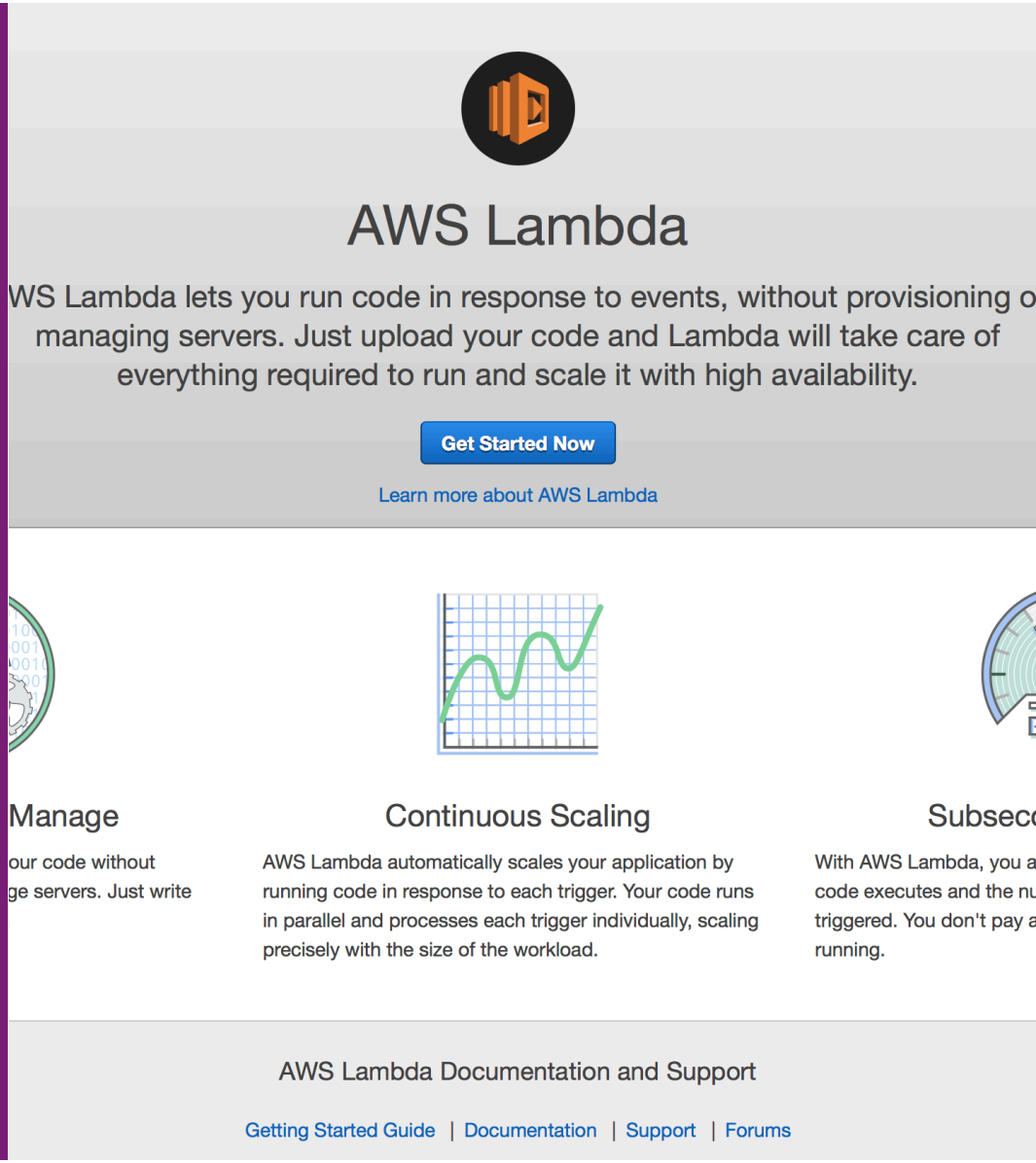
AWS LAMBDA

- The Lambda free tier includes 1M free requests per month and 400,000 GB-seconds of compute time per month
- The memory size you choose for your Lambda functions determines how long they can run in the free tier
- The Lambda free tier does not automatically expire at the end of your 12 month AWS Free Tier term, but is available forever

Memory (MB)	Free tier seconds per month	Price per 100ms ($)
128	3,200,000	0.000000208
192	2,133,333	0.000000313
256	1,600,000	0.000000417
320	1,280,000	0.000000521
384	1,066,667	0.000000625
448	914,286	0.000000729
512	800,000	0.000000834
576	711,111	0.000000938
640	640,000	0.000001042
704	581,818	0.000001146
768	533,333	0.000001250
832	492,308	0.000001354
896	457,143	0.000001459
960	426,667	0.000001563
1024	400,000	0.000001667
1088	376,471	0.000001771
1152	355,556	0.000001875
1216	336,842	0.000001980

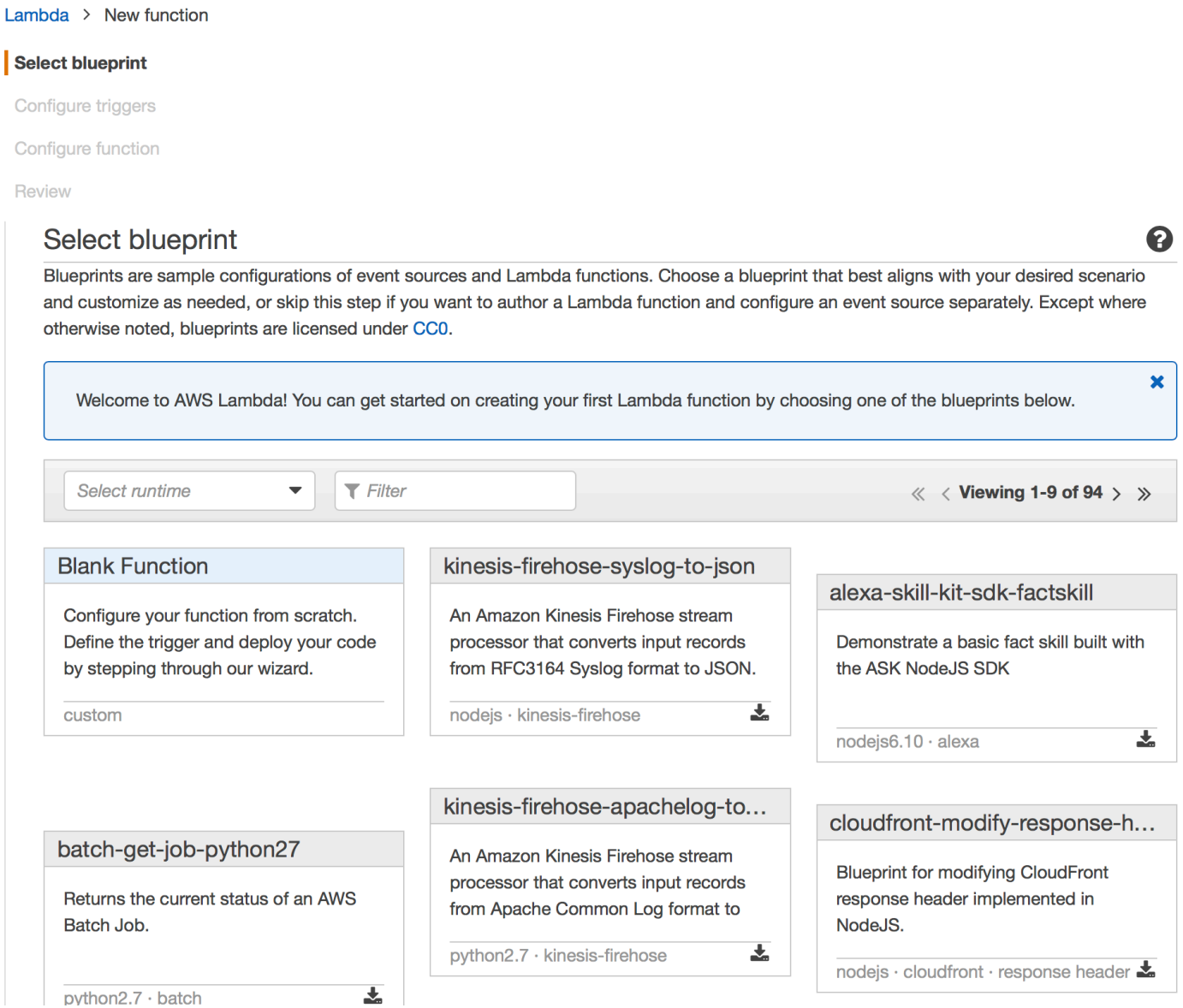
AWS LAMBDA

- Create an AWS account
- Create admin account (optional)
- Download AWS CLI
  - Homebrew or pip
- Create and deploy



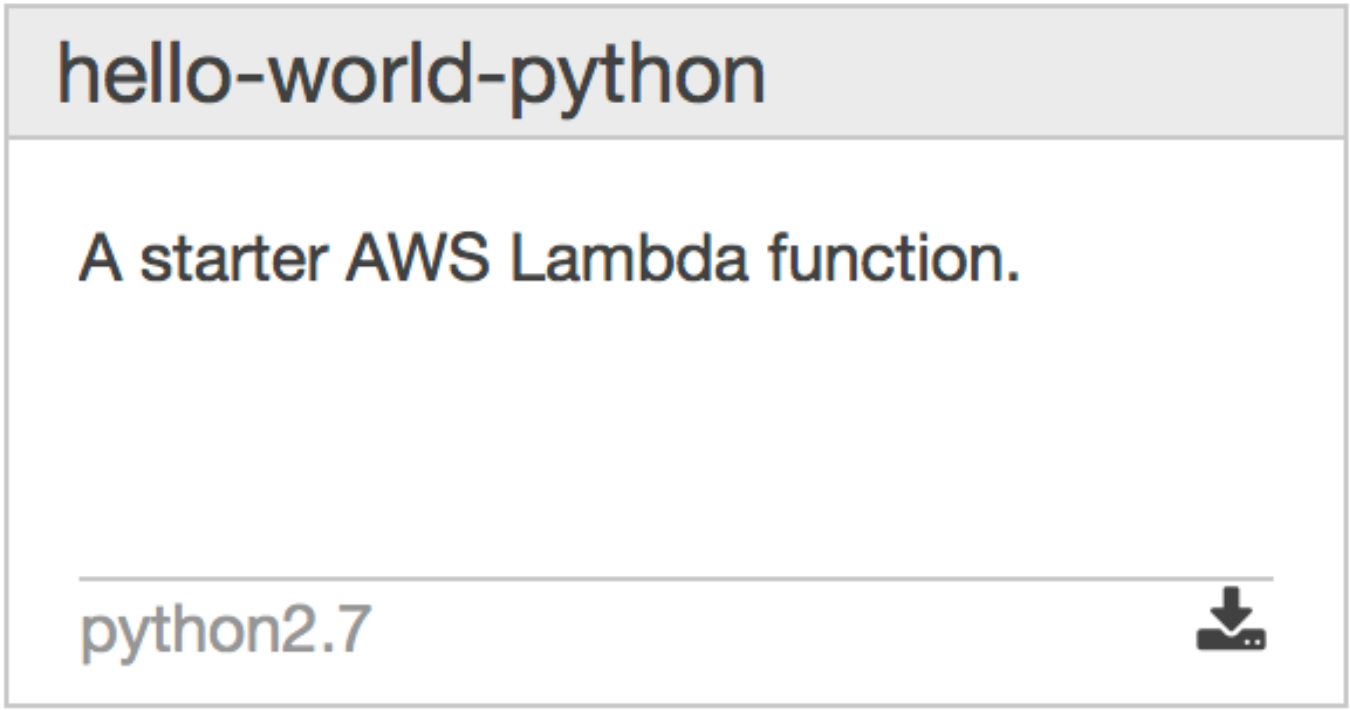
AWS LAMBDA

- Blueprints are sample configurations of event sources and Lambda functions
- Choose a blueprint that best aligns with your desired scenario and customize



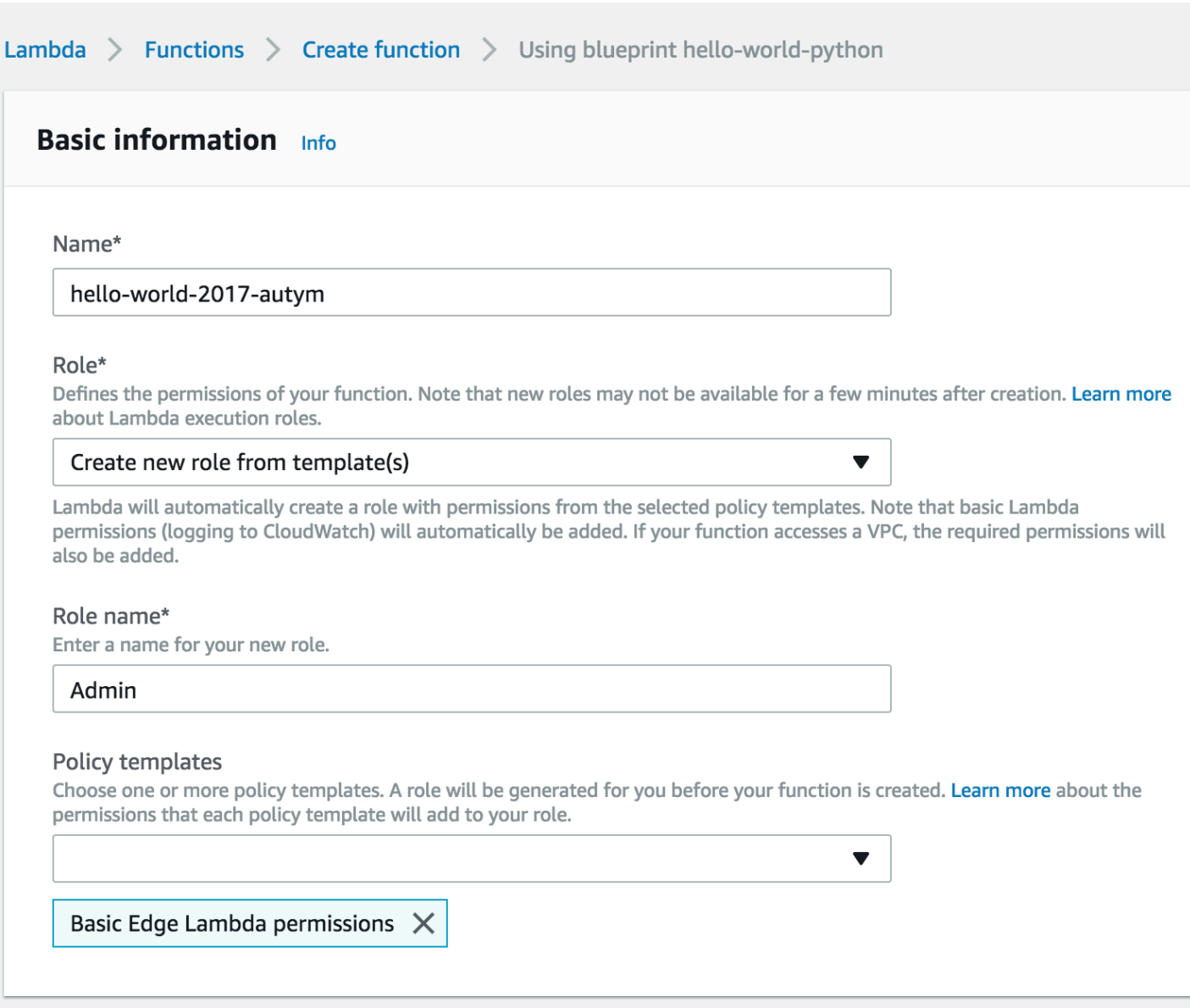
AWS LAMBDA

- Warning: the configurations for the blueprints don't all work out of the box



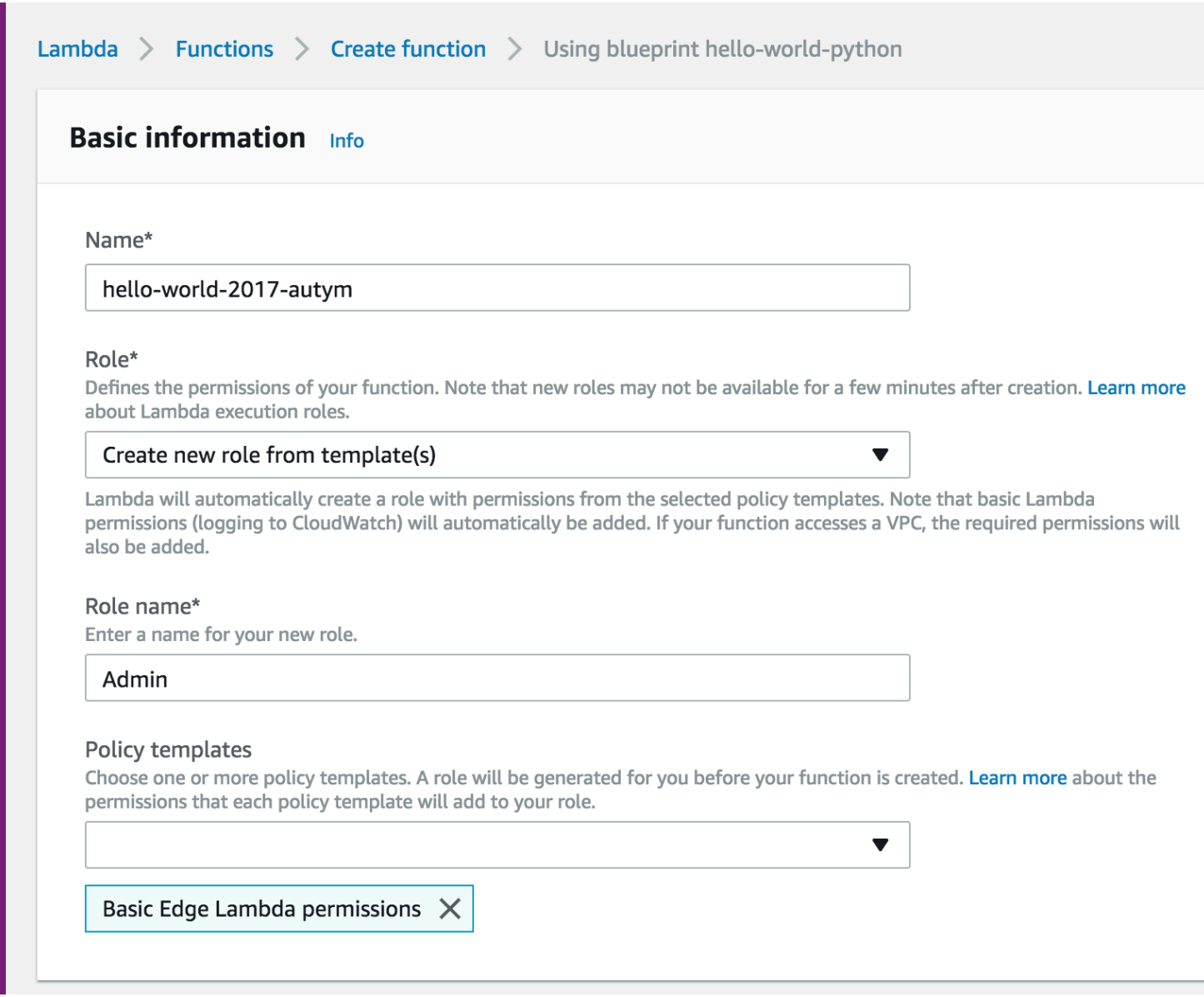
AWS LAMBDA

- Create functions
  - In-browser editor
  - External gzip and push



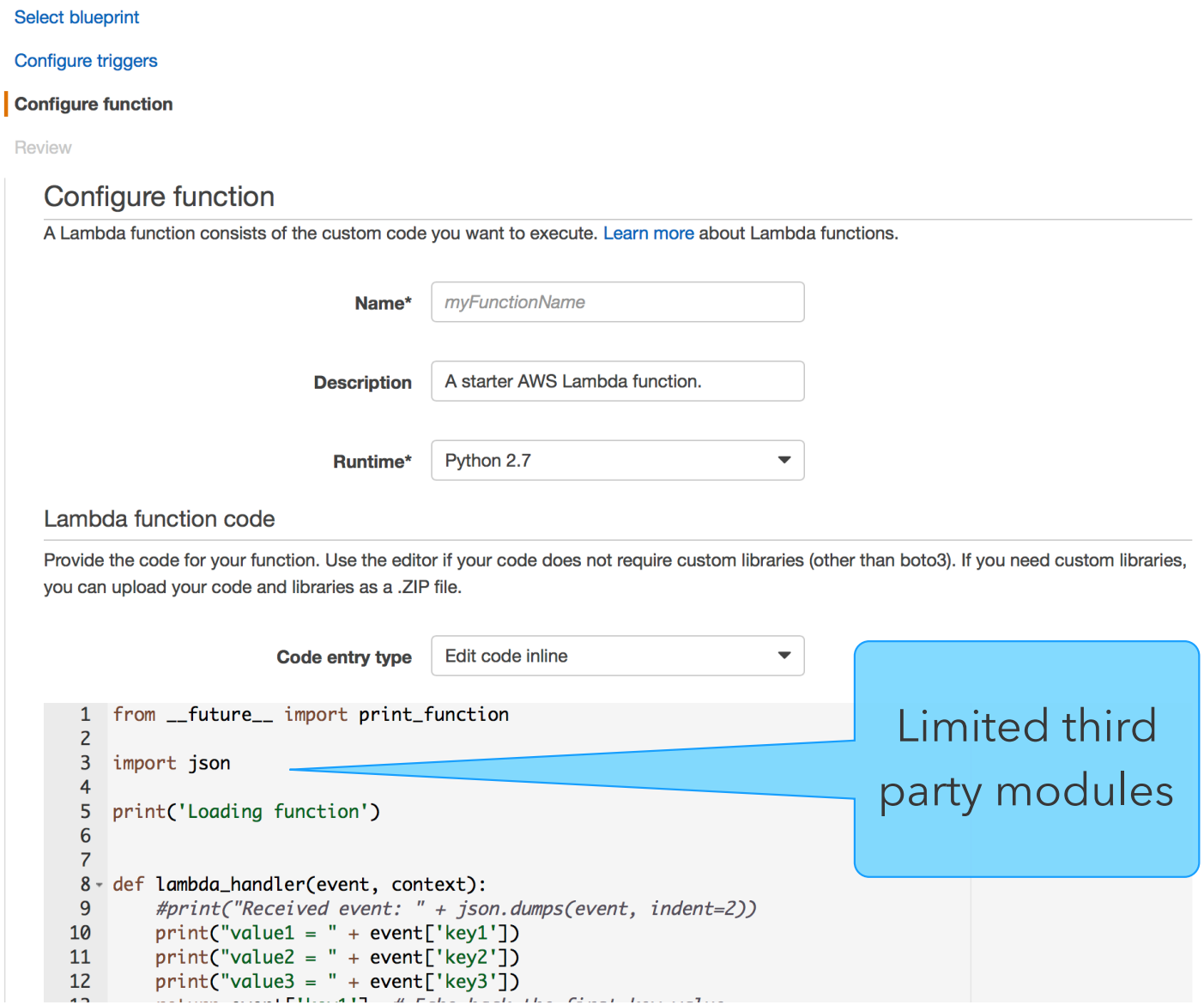
AWS LAMBDA

- Create functions
  - Set roles
  - Set permissions



AWS LAMBDA

- Functions can be written in JS, Python, Java, and Go



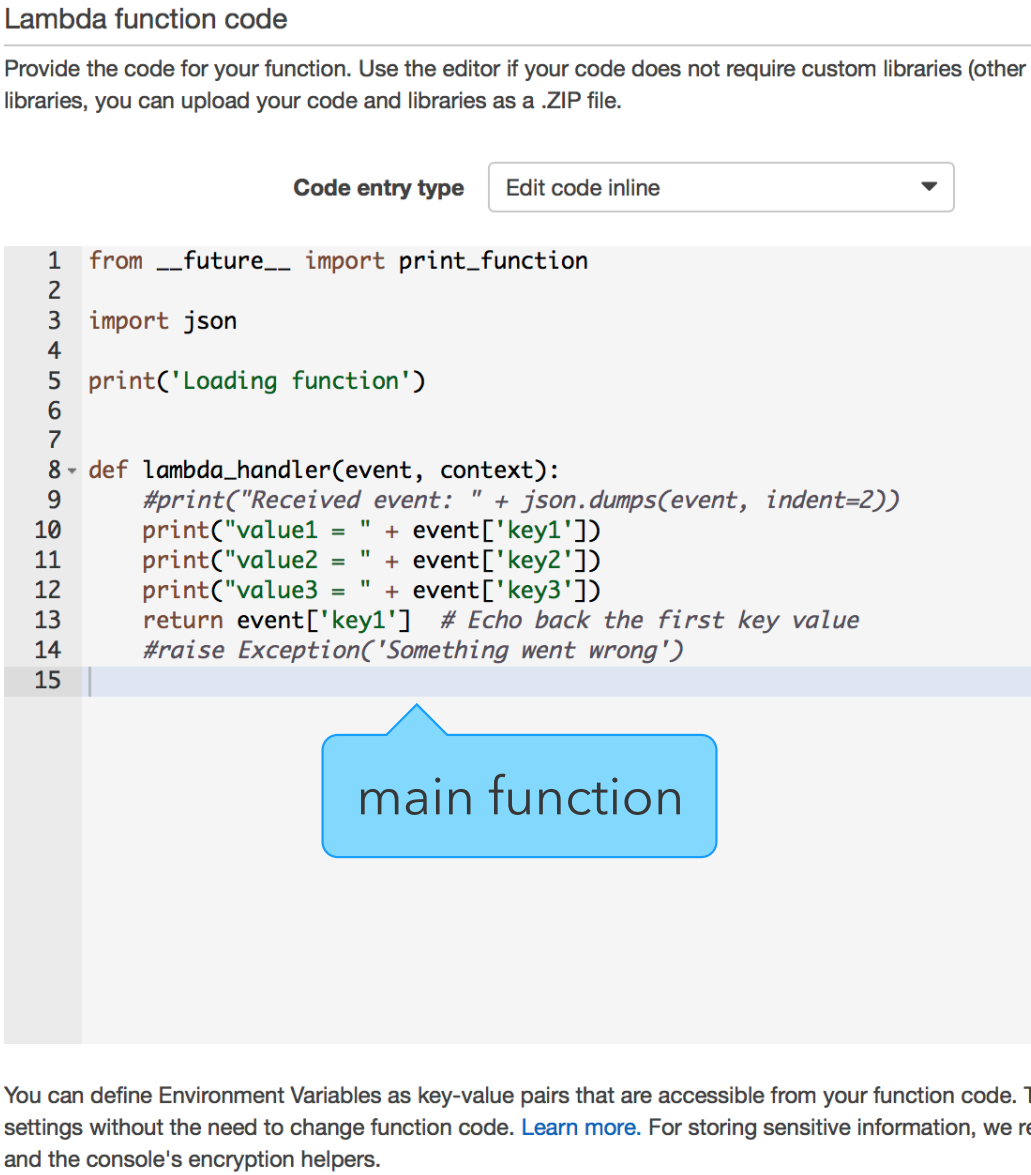
AWS LAMBDA

- Triggers define when you function runs
  - Event based to other AWS Services (push to S3; filter photo)
  - HTTP API
  - Mobile API
  - Scheduled Events (cron)

Functions can be triggered by different events

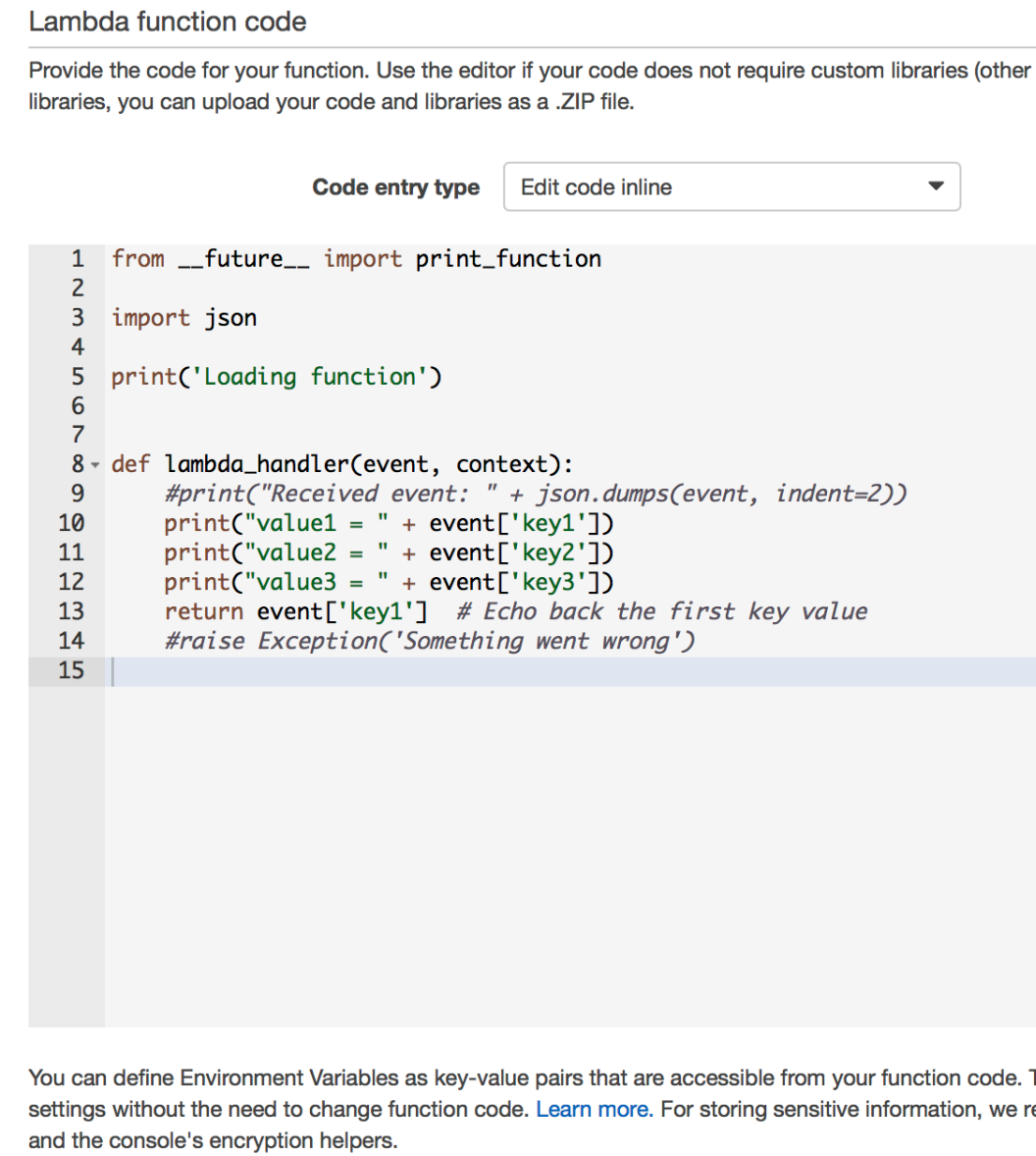
AWS LAMBDA

- Function handler is the main function called
  - `lambda_function.lambda_handler` is the default console function
  - Name doesn't matter



AWS LAMBDA

- Parameters
  - Event - data passed in
  - Context - runtime information



AWS LAMBDA

- Return values depends on how you set up the function
- Functions need to return properly to test

unction.py

```
from __future__ import print_function

import json

print('Loading function')


def lambda_handler(event, context):
    return {"statusCode": 200,  \
        "headers": {"Content-type": "application/json"}, \
        "body": "{\"count\": 5, \"message\": \"hello\"}", \
        "isBase64Encoded": "false"}
```

AWS LAMBDA

▾ Advanced settings

These settings allow you to control the code execution performance and costs for your Lambda function. Changing your resource settings (by selecting memory) or changing the timeout may impact your function cost. Learn more about how Lambda pricing works.

Memory (MB)* 128

Timeout* 0 min 3 sec

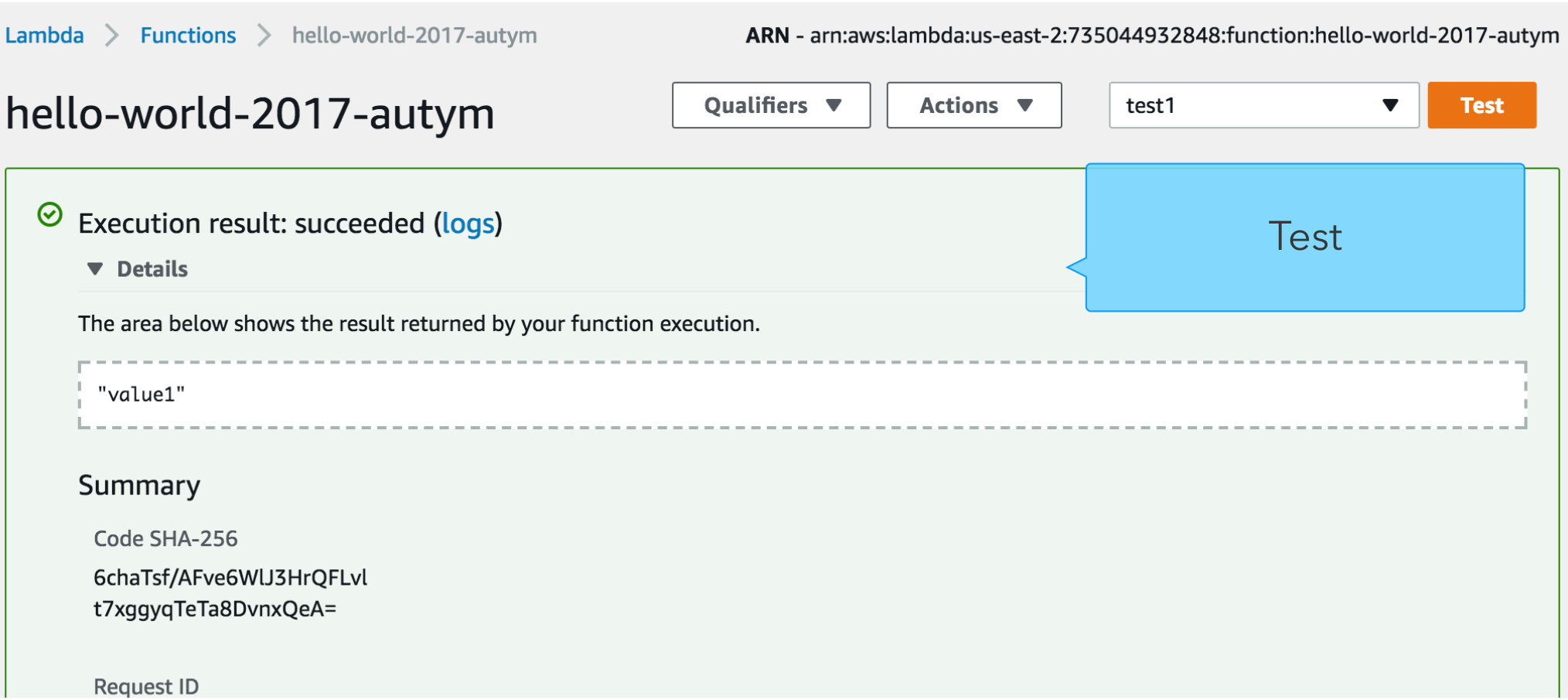
Sufficient for Hello World

AWS Lambda will automatically retry failed executions for asynchronous invocations. You can additionally optionally configure Lambda to forward payloads that were not processed to a dead-letter queue (DLQ), such as an SQS queue or an SNS topic. Learn more about Lambda's retry policy and DLQs. **Please ensure your role has appropriate permissions to access the DLQ resource.**

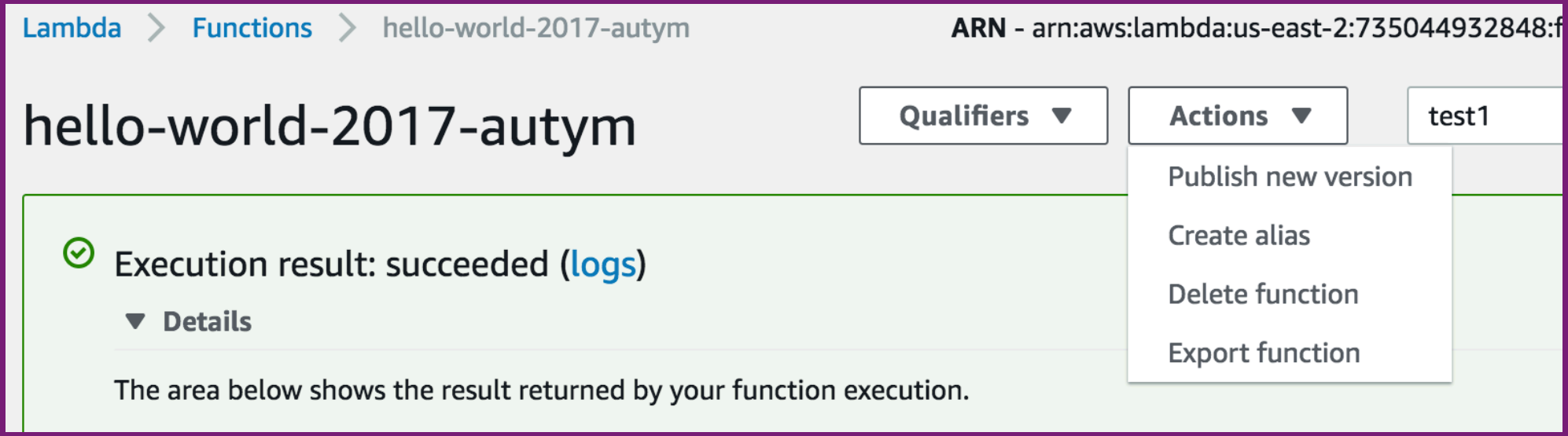
DLQ Resource Select resource

All AWS Lambda functions run securely inside a default system-managed VPC. However, you can optionally configure Lambda to access resources, such as databases, within your custom VPC. Learn more about accessing VPCs within Lambda. **Please ensure your role has**

AWS LAMBDA


Lambda > Functions > hello-world-2017-autym
ARN - arn:aws:lambda:us-east-2:735044932848:function:hello-world-2017-autym
hello-world-2017-autym
Qualifiers
Actions
test1
Test
Execution result: succeeded (logs)
Details
The area below shows the result returned by your function execution.
"value1"
Summary
Code SHA-256
6chaTsf/AFve6WlJ3HrQFLvl
t7xggyqTeTa8DvnxQeA=
Request ID
Test

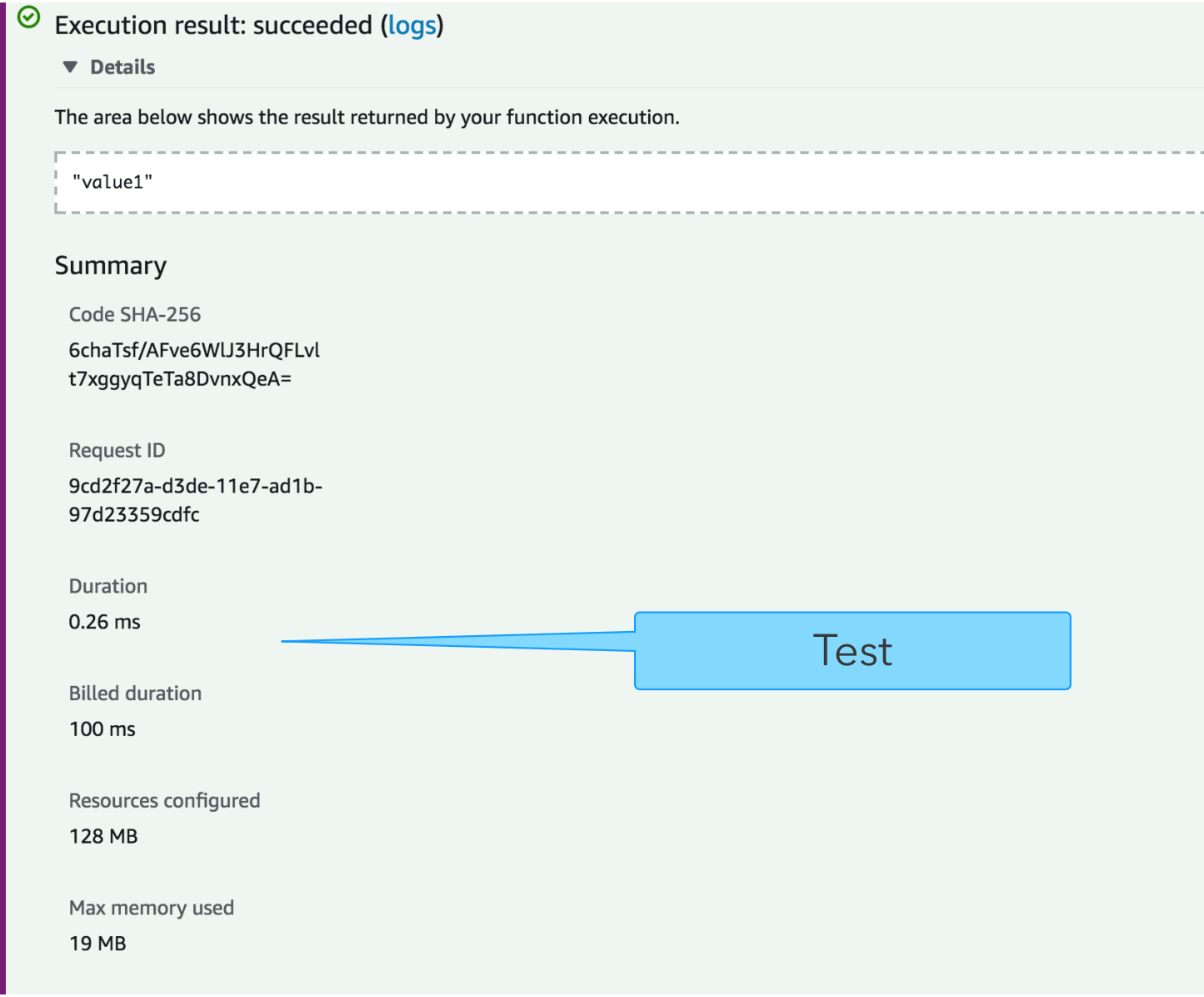
AWS LAMBDA



- Review your Lambda function before deploying
- You can test and edit

AWS LAMBDA

- Test function
  - Response
  - Log output
  - Runtime summary



AWS LAMBDA

Print statements go to logging

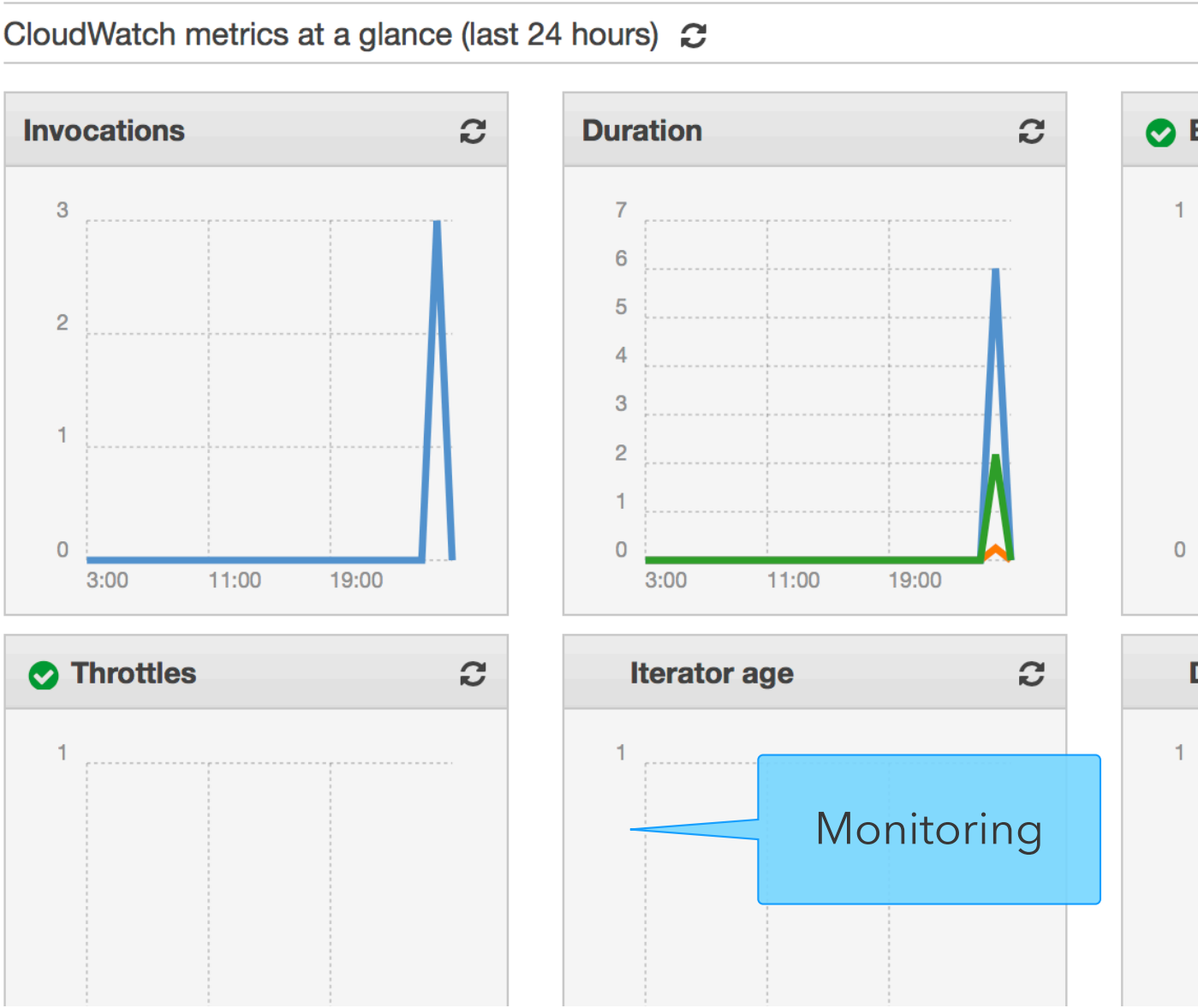
Log output

The area below shows the logging calls in your code. These correspond to a single row within the CloudWatch log group corresponding to this Lambda function. Click here to view the CloudWatch log group.

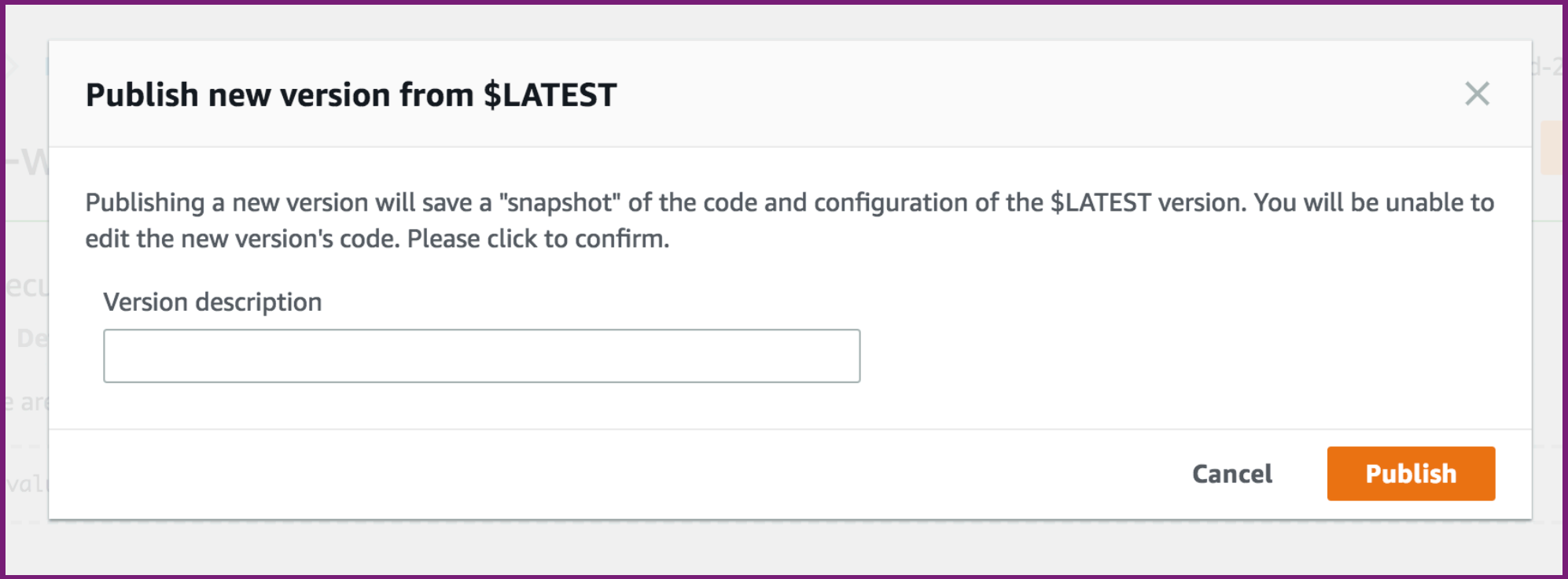
```
START RequestId: 9cd2f27a-d3de-11e7-ad1b-97d23359cdfc Version: $LATEST
value1 = value1
value2 = value2
value3 = value3
END RequestId: 9cd2f27a-d3de-11e7-ad1b-97d23359cdfc
REPORT RequestId: 9cd2f27a-d3de-11e7-ad1b-97d23359cdfc  Duration: 0.26 ms       Billed Duration: 100 ms         Memory Size:
128 MB  Max Memory Used: 19 MB
```

AWS LAMBDA

- Monitor you function



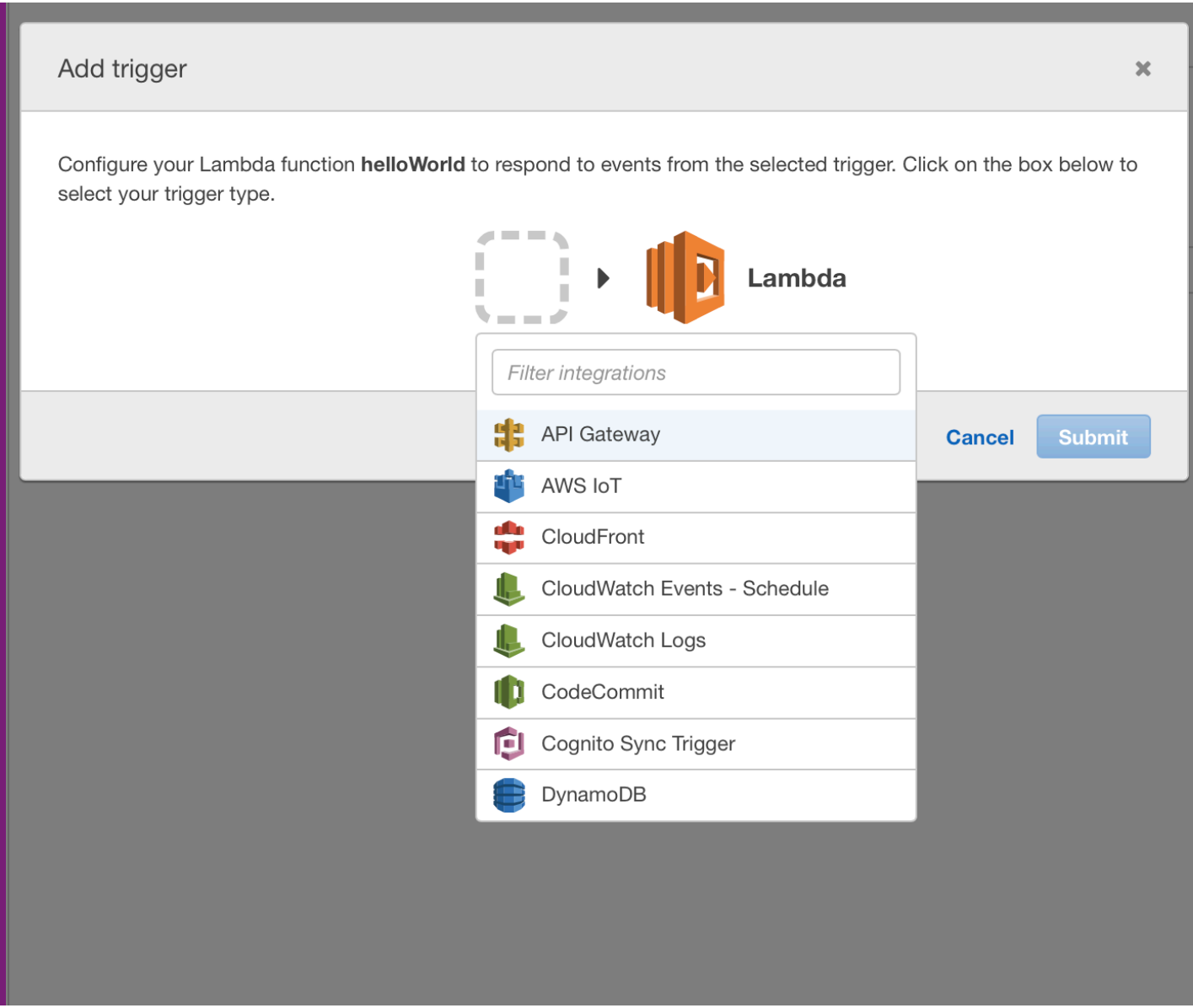
AWS LAMBDA



- Publish

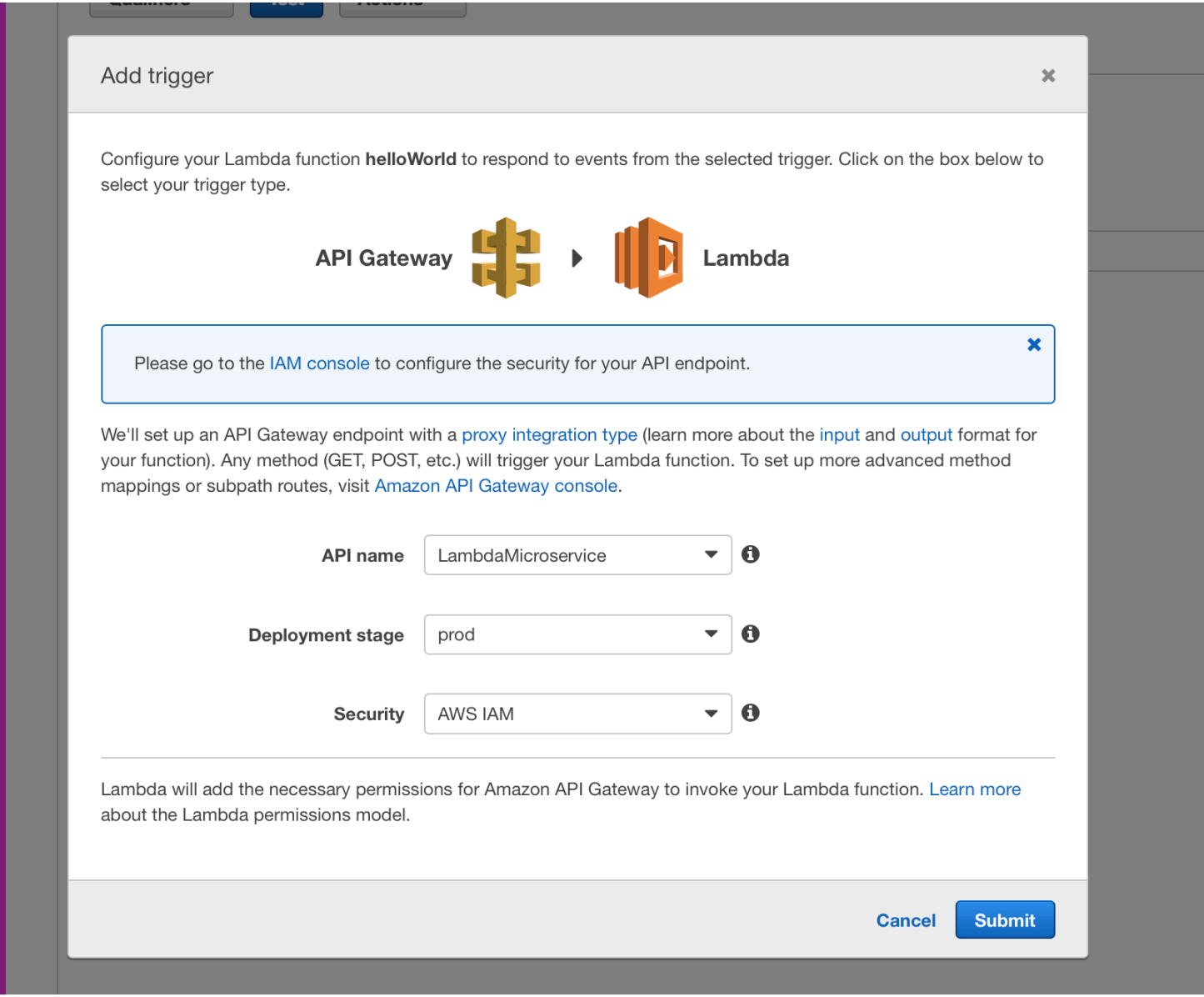
AWS LAMBDA

- Set up a trigger

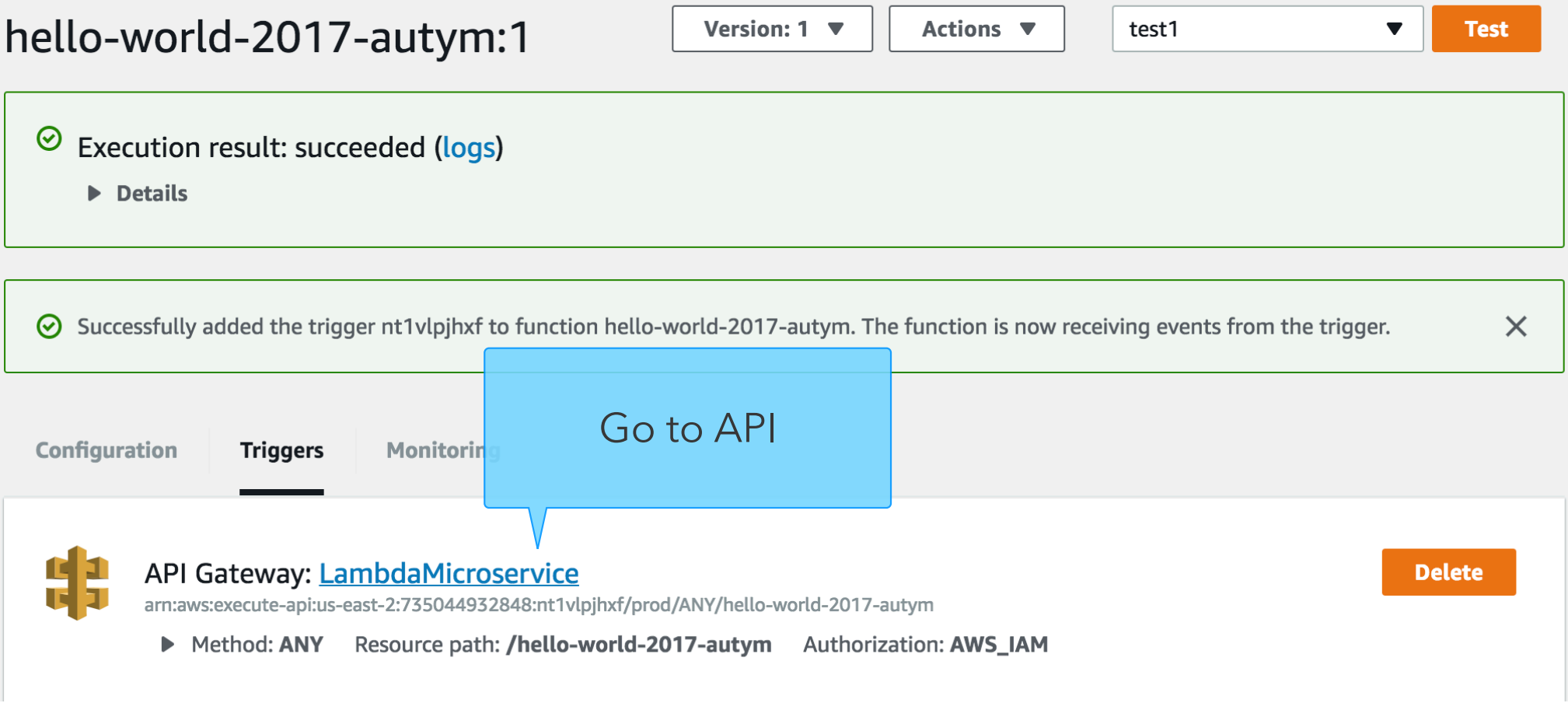


AWS LAMBDA

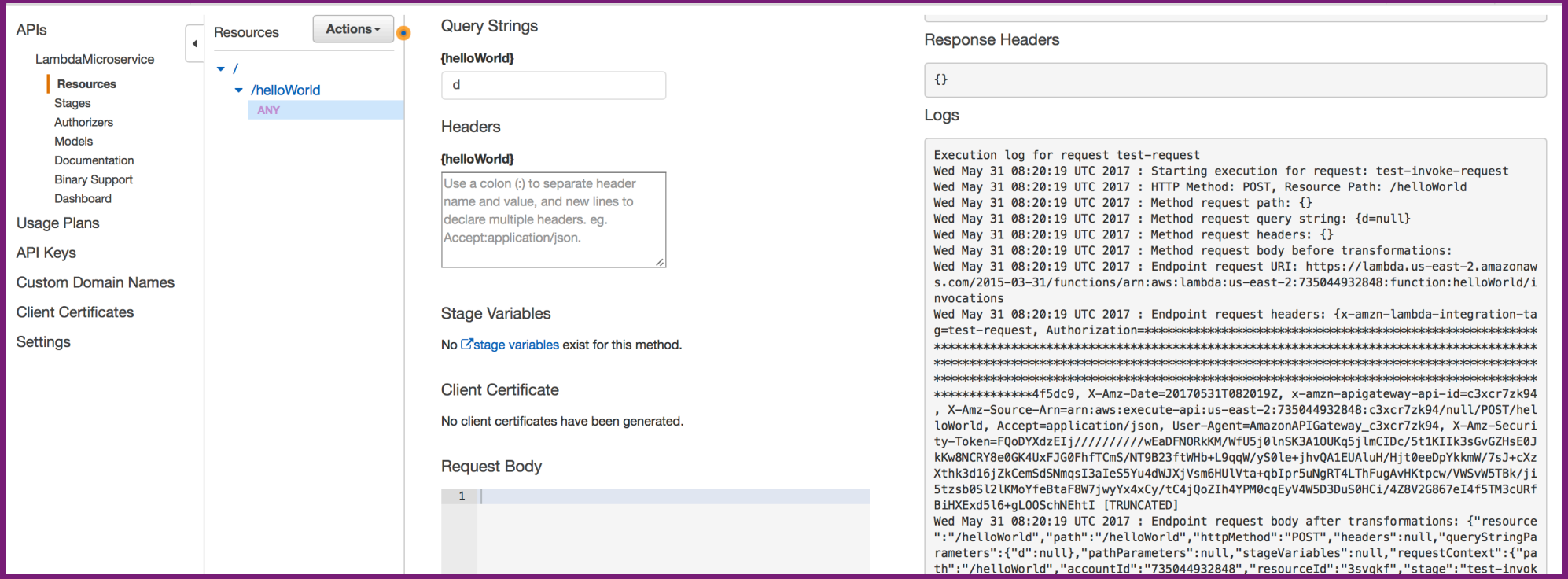
- More security hoops to jump through
- In my experience, you may need to delete the trigger after updating



AWS LAMBDA


hello-world-2017-autym:1
Version: 1
Actions
test1
Test
Execution result: succeeded (logs)
Details
Successfully added the trigger nt1vlpjhxf to function hello-world-2017-autym. The function is now receiving events from the trigger.
Configuration
Triggers
Monitoring
Go to API
API Gateway: LambdaMicroservice
arn:aws:execute-api:us-east-2:735044932848:nt1vlpjhxf/prod/ANY/hello-world-2017-autym
Delete
Method: ANY
Resource path: /hello-world-2017-autym
Authorization: AWS_IAM

AWS LAMBDA



- Test your API (note that to be successful, need to return proper code)

AWS LAMBDA

```
import json

print('Loading function')


def lambda_handler(event, context):
    return {"statusCode": 200,  \
        "headers": {"Content-type": "application/json"}, \
        "body": "{\"count\": 5, \"message\": \"hello\"}", \
        "isBase64Encoded": "false"}
```

- Test your API (note that to be successful, need to return proper code)

AWS LAMBDA

- CloudWatch trigger
  - Schedule
  - AWS Event

Rule
Pick an existing rule, or create a new one.

Create a new rule

Select or create a new rule

Rule name*
Enter a name to uniquely identify your rule.

Rule description
Provide an optional description for your rule.

Rule type
Trigger your target based on an event pattern, or based on an automated schedule.

- ○ Event pattern
- ◉ Schedule expression

Schedule expression*
Self-trigger your target on an automated schedule using Cron or rate expressions. Cron expressions are in UTC.

e.g. rate(1 day), cron(0 17 ? * MON-FRI *)

AWS LAMBDA

- Limits to consider for lambda

AWS Lambda Resource Limits

Resource	Default Limit
Ephemeral disk capacity ("/tmp" space)	512 MB
Number of file descriptors	1,024
Number of processes and threads (combined total)	1,024
Maximum execution duration per request	300 seconds
Invoke request body payload size (RequestResponse)	6 MB
Invoke request body payload size (Event)	128 K
Invoke response body payload size (RequestResponse)	6 MB

The following table lists the Lambda account limits per region.

AWS Lambda Account Limits Per Region

Resource	Default Limit
Concurrent executions (see Lambda Function Concurrent Executions).	1000

FIREBASE CLOUD FUNCTIONS VS LAMBDA

🔥FIREBASE CLOUD FUNCTIONS

- https://youtu.be/vr0Gfvp5v1A

🔥FIREBASE CLOUD FUNCTIONS

- Integrates with Firebase in different ways
  - Realtime Database Triggers
  - Firebase Authentication Triggers
  - Firebase Analytics Triggers
  - Cloud Storage Triggers
  - Cloud Pub/Sub Triggers
  - HTTP Triggers

🔥FIREBASE
CLOUD FUNCTIONS

- Overall the functionality and offerings are comparable
  - Deploy functions in the cloud
  - Integrations with other offering (aws and Google ecosystem)
- Devil is in the details

CREATING FUNCTIONS

CREATING FUNCTIONS

- Lambda
  - Nice testing environment
  - In-browser editor
  - Language support
  - More trigger variety (cron)
  - Ton of boilerplate code

CREATING FUNCTIONS

- Firebase
  - Limited (now) to JS
  - Functions can be tied to other Firebase/Google events
  - Fewer triggers

The functions you write can respond to events generated by these other Firebase and Google Cloud features:

- Realtime Database Triggers
- Firebase Authentication Triggers
- Google Analytics for Firebase Triggers
- Cloud Storage Triggers
- Cloud Pub/Sub Triggers
- HTTP Triggers

DEPLOYING

DEPLOYING

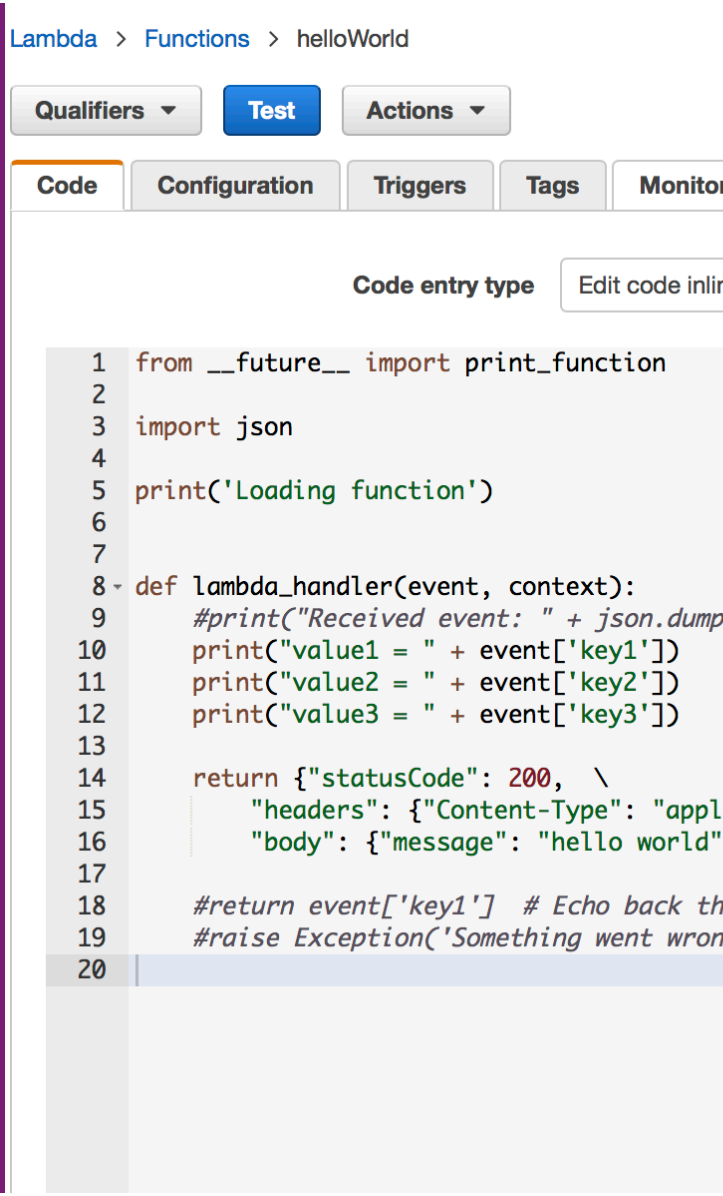
- Deployment and setup is similar
- Lamda does have the in-brower development and deployment

```
aws deploy

firebase deploy
```

TESTING

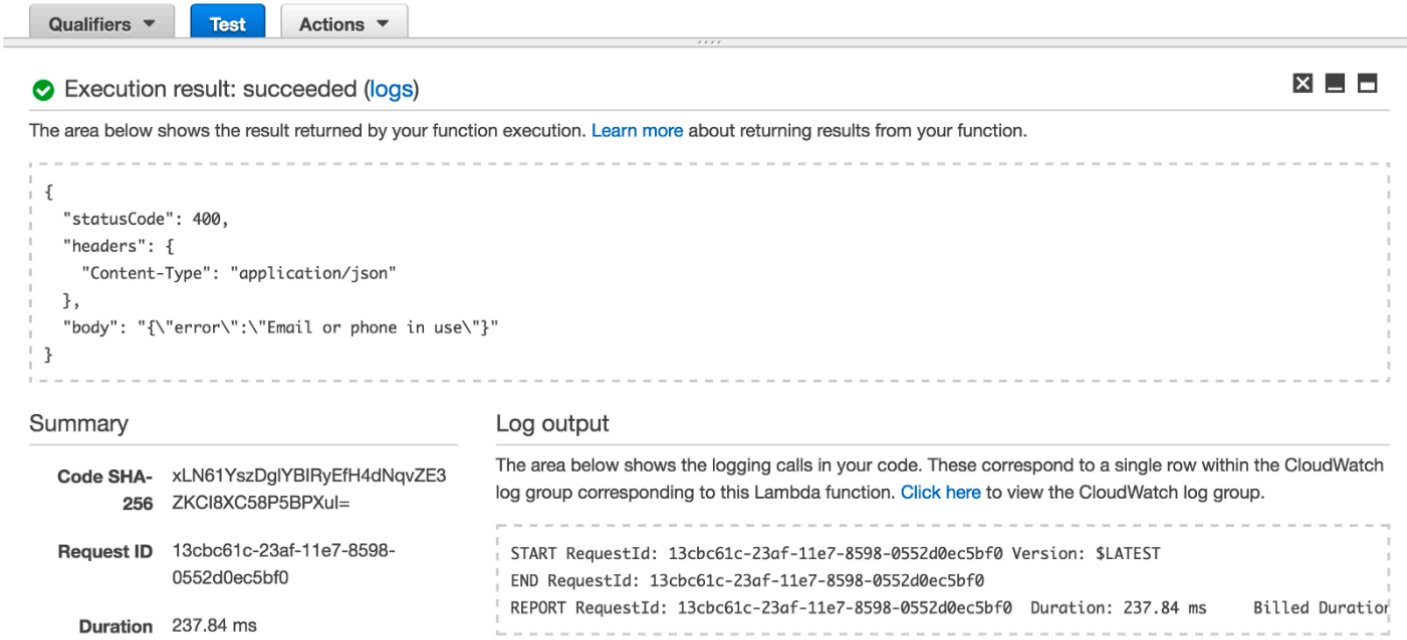
TESTING

- Lambda
  - Great testing environment
  - Test functions and API



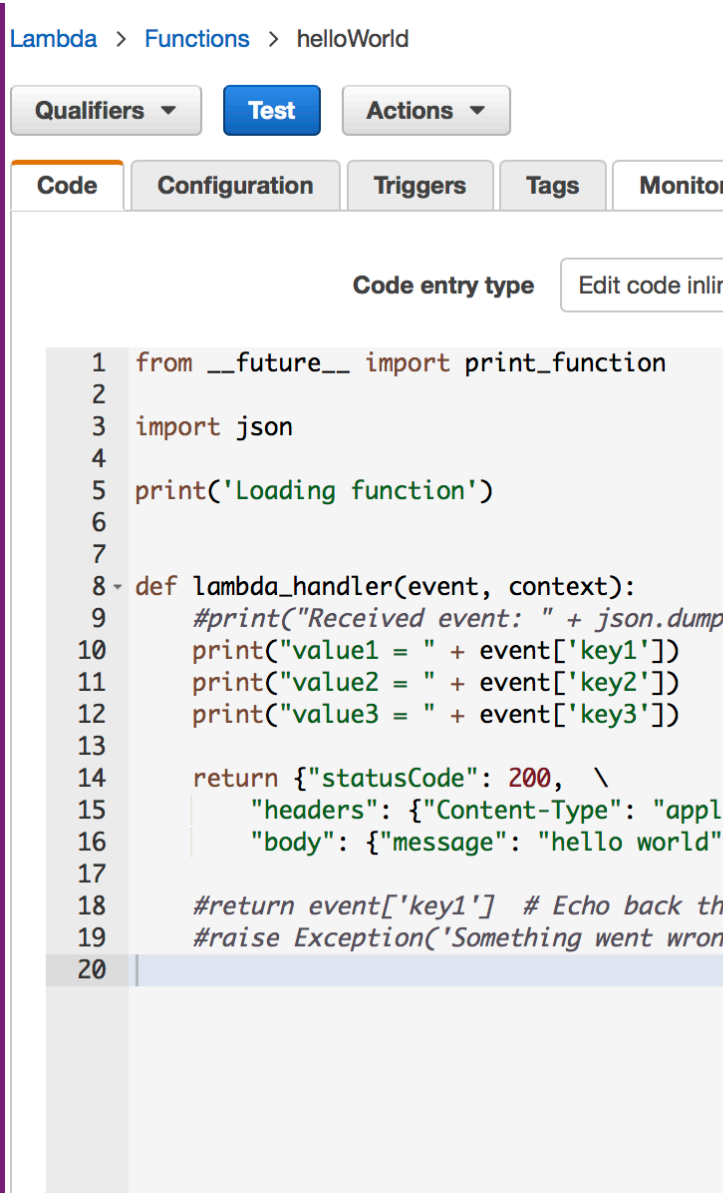
TESTING

- Lambda
  - Debug console and logging provides useful information



TESTING

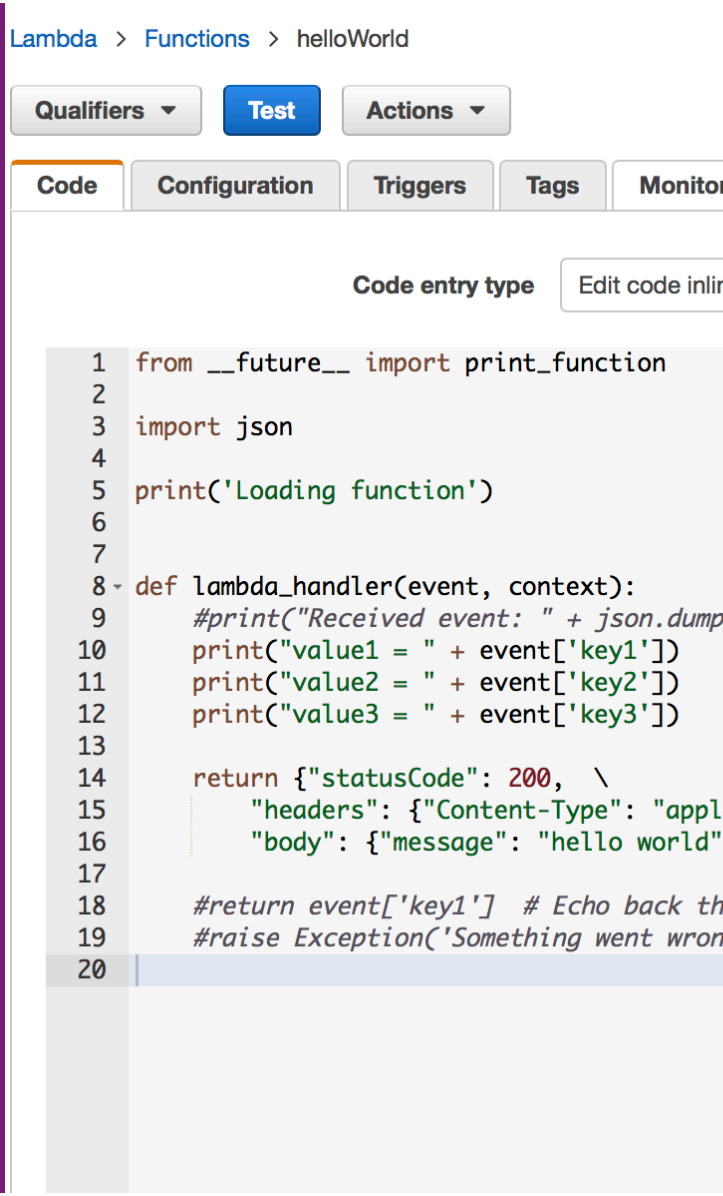
- Firebase
  - Local development server
  - Deploy/test cycle is slow
  - Console only shows logged information



PRICING

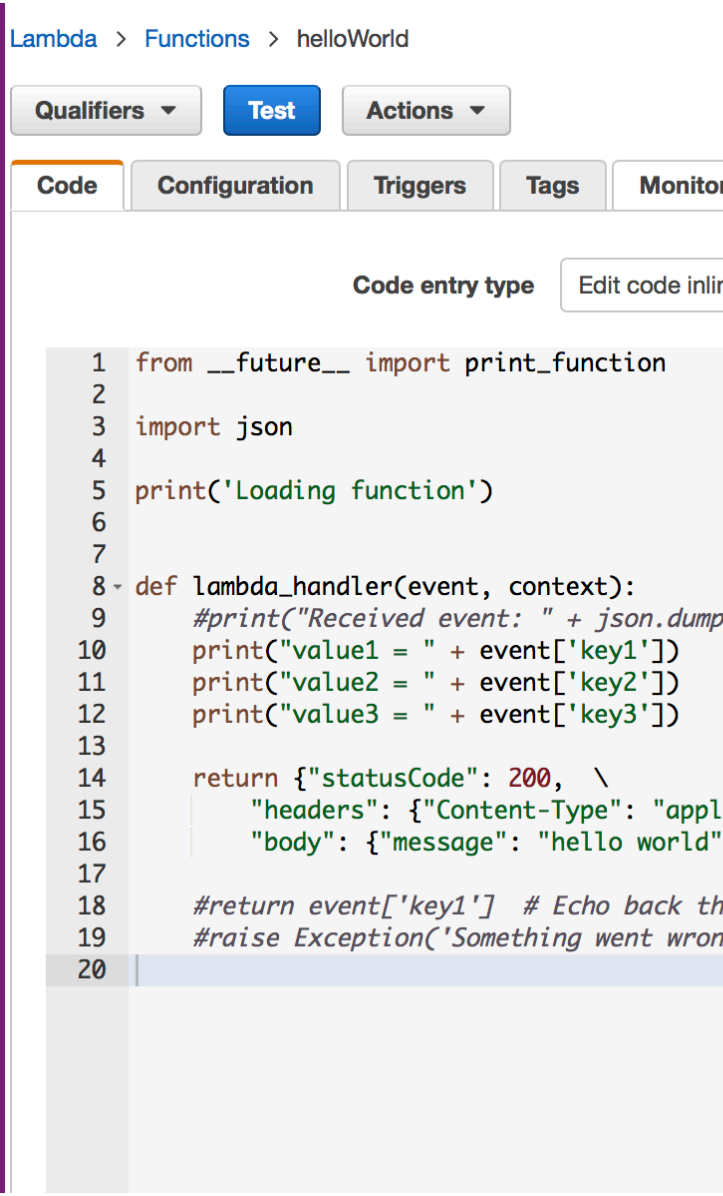
PRICING

- Pricing metrics
  - Number of invocations
  - Duration of invocation
  - Hardware



PRICING

- Cloud Functions
  - $.40 per million (2M free)
- Lambda
  - $.20 per millions (1M free)



ODDS AND ENDS

STATUS AND DOCUMENTATION

- Cloud functions
  - In beta
  - Up and coming
  - Updated documentation

CLOUD FUNCTIONS BETA

A serverless environment to build and connect cloud services

VIEW DOCUMENTATION VIEW CONSOLE

Serverless Applications on Google's Infrastructure

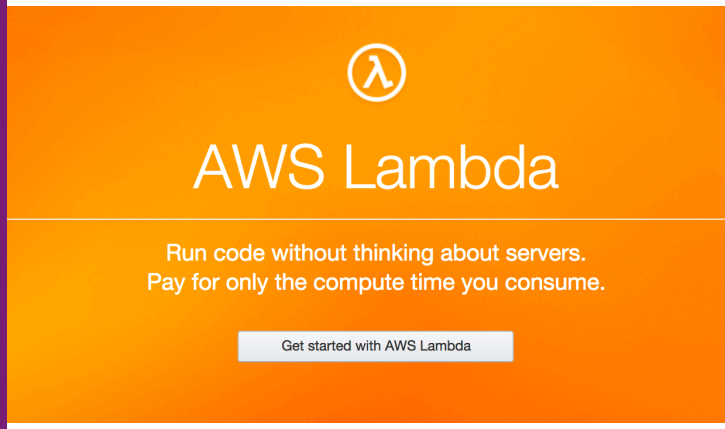
Cloud computing has made possible fully serverless models of computing where logic can be spun up on-de
in response to events originating from anywhere. Construct applications from bite-sized business logic billed
the nearest 100 milliseconds, only while your code is running. Serve users from zero to planet-scale, all witho
managing any infrastructure.

Microservices Over Monolith

Developer agility comes from building syst
functionality focused on doing one thing w
services at the level of a single function, n
VMs.

STATUS AND DOCUMENTATION

- Lambda
  - Generally available
  - Documentation is outdated in many circumstances
  - Stack Overflow has correct answers to old problems



STATUS AND DOCUMENTATION

- App Engine
  - More control
  - 3rd party modules
  - Any language

GOOGLE APP ENGINE

Build scalable web and mobile backends in any language on Google's infrastructure

VIEW DOCUMENTATION

App Engine for All

Build modern web and mobile applications on an open cloud platform: bring your own langu
runtimes, frameworks, and third party libraries. Google App Engine is a fully managed platfo
completely abstracts away infrastructure so you focus only on code. Go from zero to planet-
see why some of today's most successful companies power their applications on App Engir

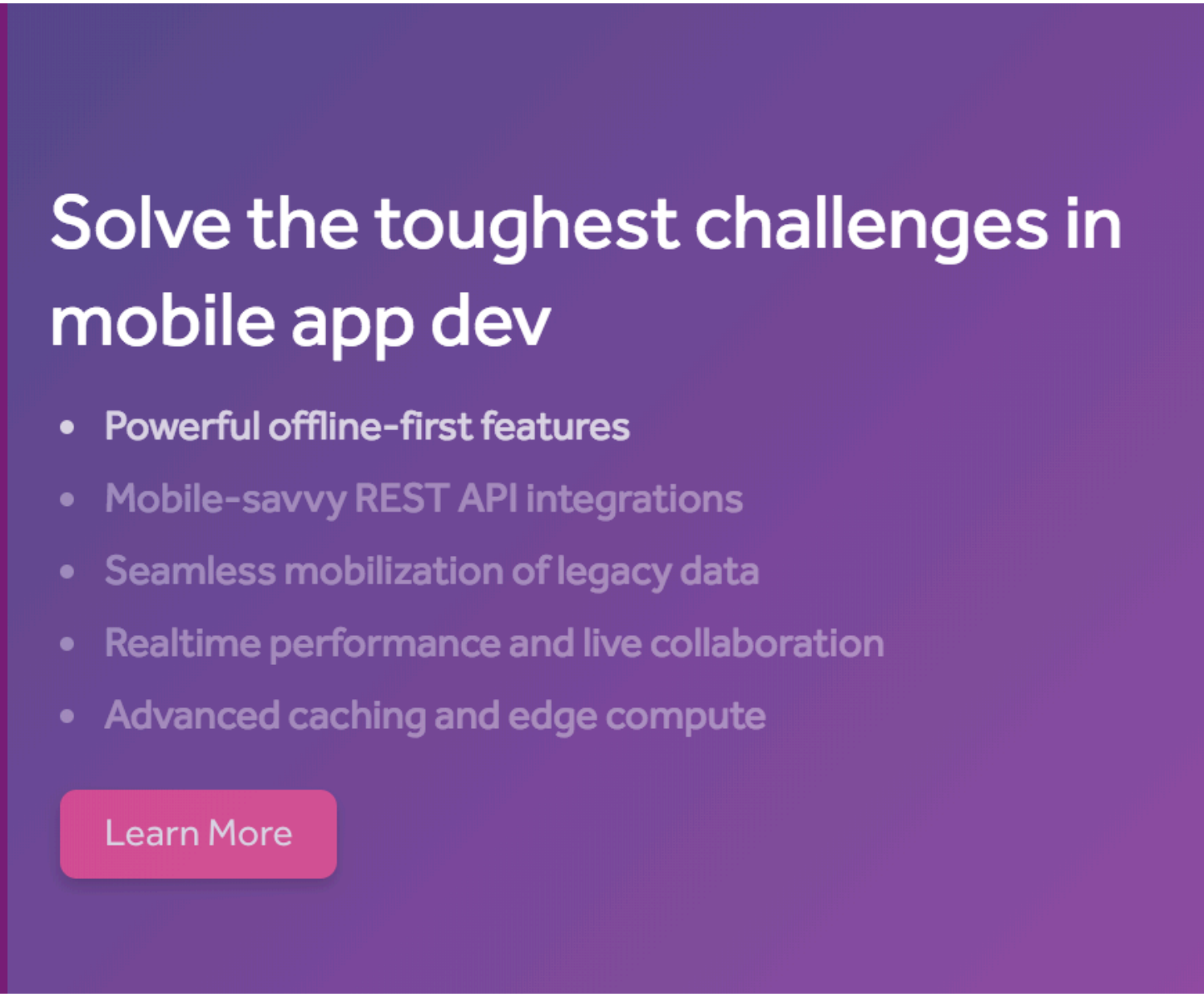
For All Language Communities

Out of the box, App Engine supports Node.js, Java, Ruby, C#, Go, Python, a
from these language communities can be productive immediately in a fam
add code.

REALM

REALM

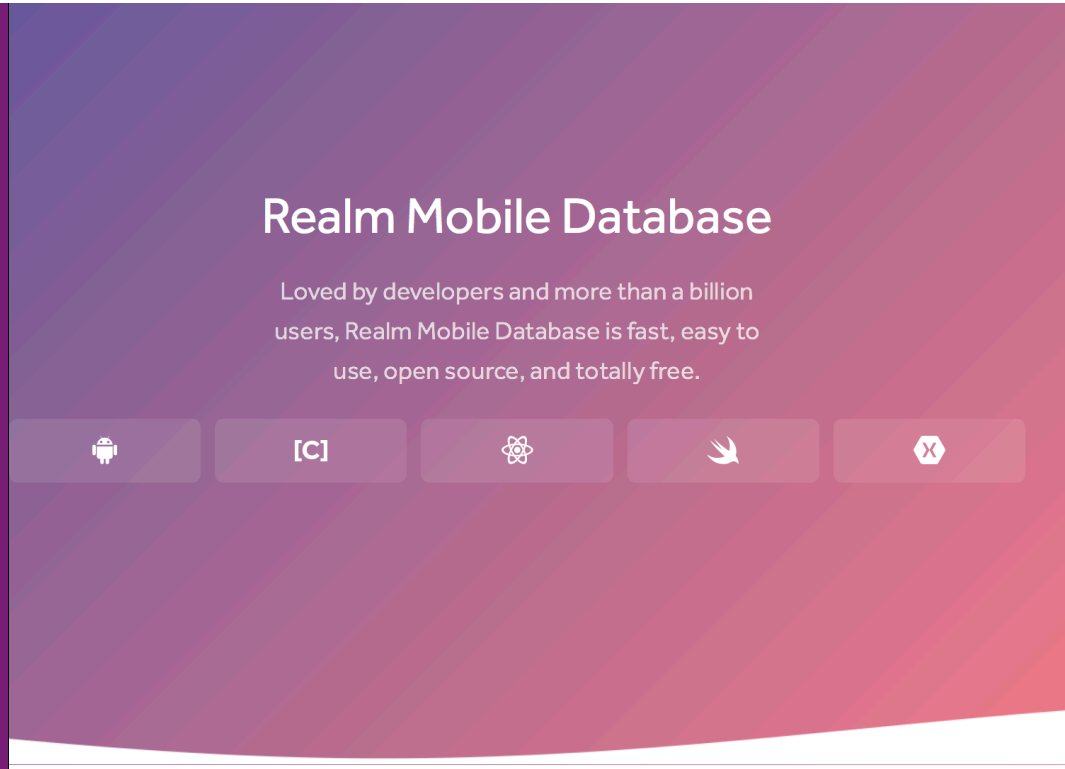
- Mobile first database
- Cross platform
- Full "Platform" for mobile app backends



IN THE BEGINNING...

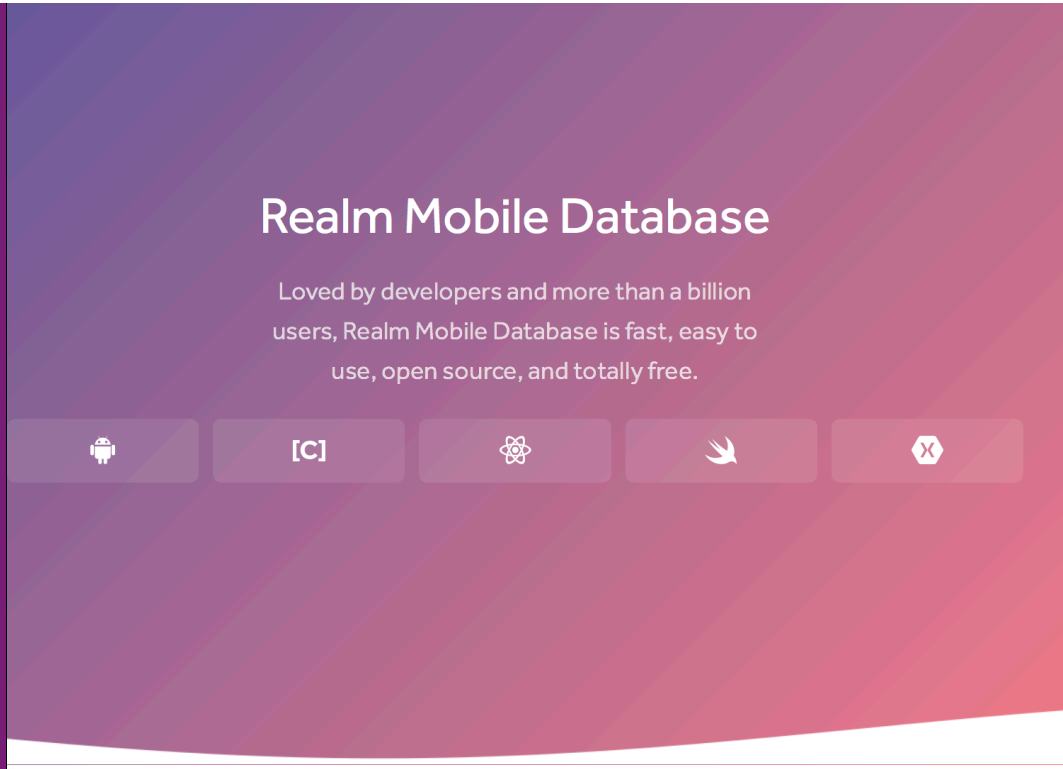
REALM

- Initially on device only
- Billed as a CoreData alternative
  - But easy to use
- Faster than SQLite



REALM

- UI Components tuned for the Realm database
  - UITableViewController/ NSFetchedResultsController



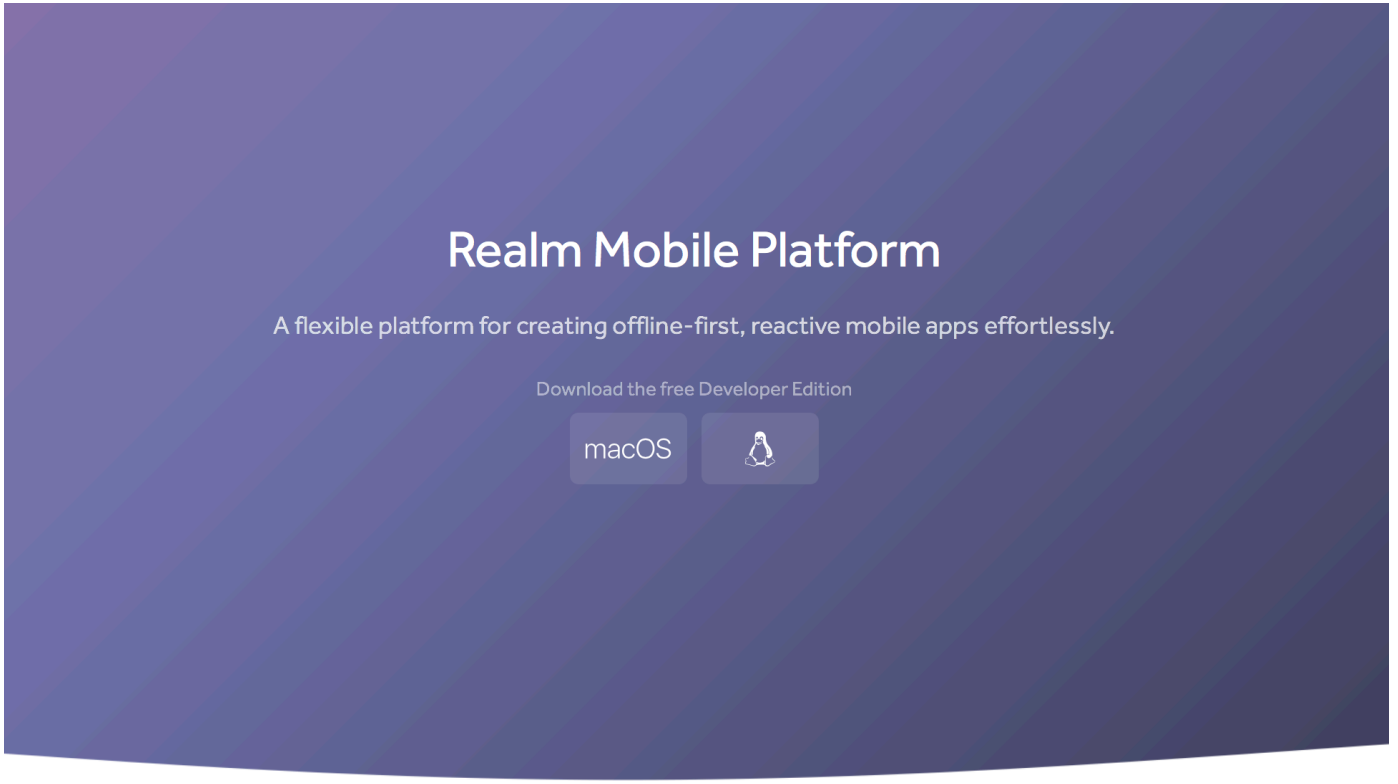
REALM

- Simple **local** object persistence
- Thread safe
- Encryption built in

```
class Dog: Object {
  dynamic var name = ""
  dynamic var age = 0
}

let dog = Dog()
dog.name = "Rex"
dog.age = 1

let realm = try! Realm()
try! realm.write {
  realm.add(dog)
}

let pups = realm.objects(Dog.self).filter("age < 2")
```

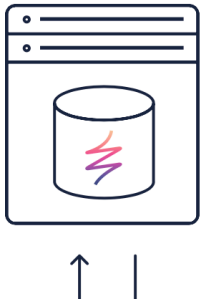
REALM

- On-Device limitation until...



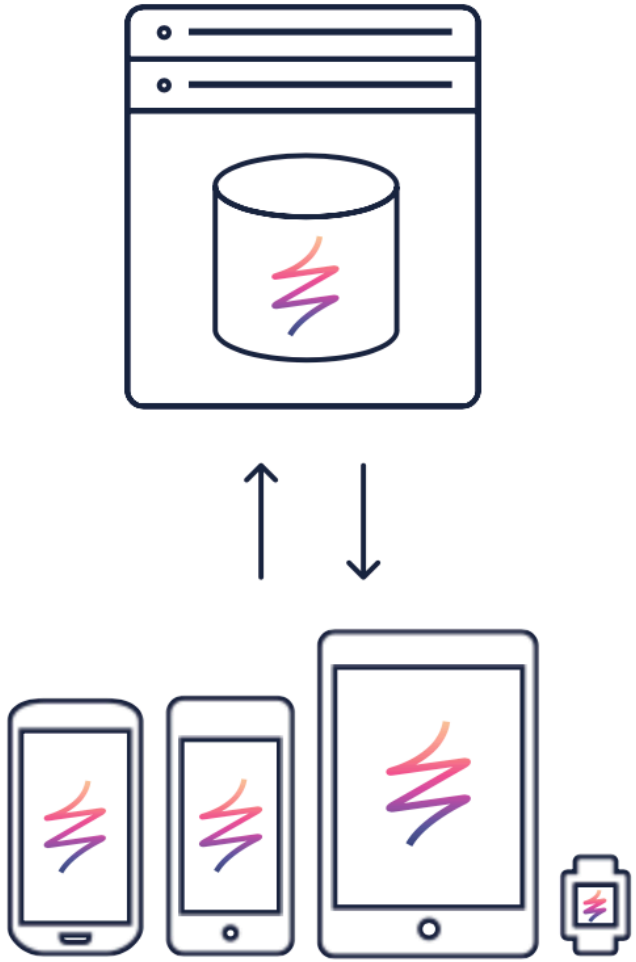
The perfect backend for the next generation of reactive mobile apps

The Realm Mobile Platform delivers automatic and seamless realtime data sync and powerful event handling between server and devices. You never need to think



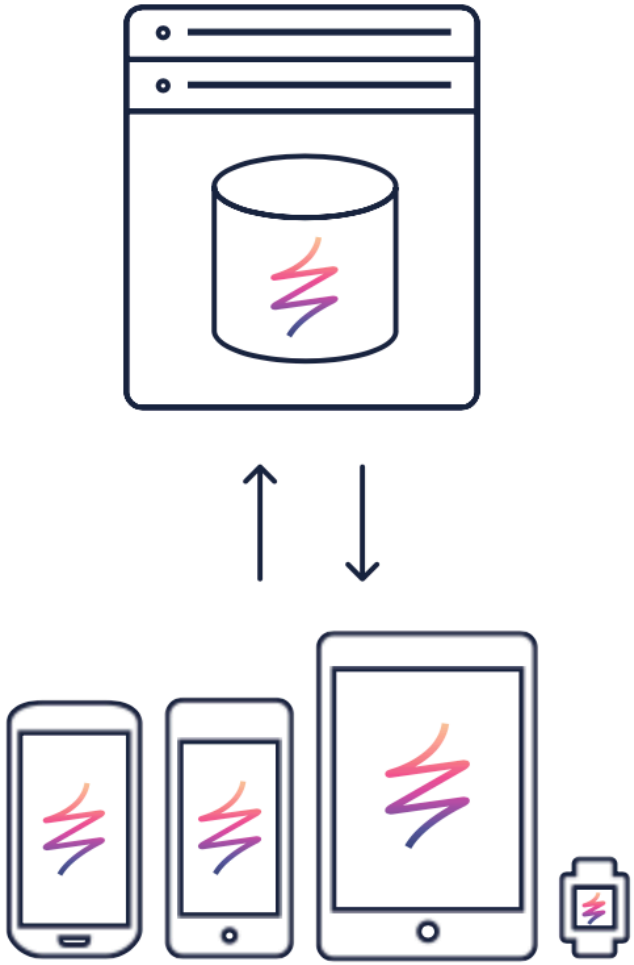
REALM

- Sync engine that runs on any cloud
- Nginx node.js server
- Public or private clouds



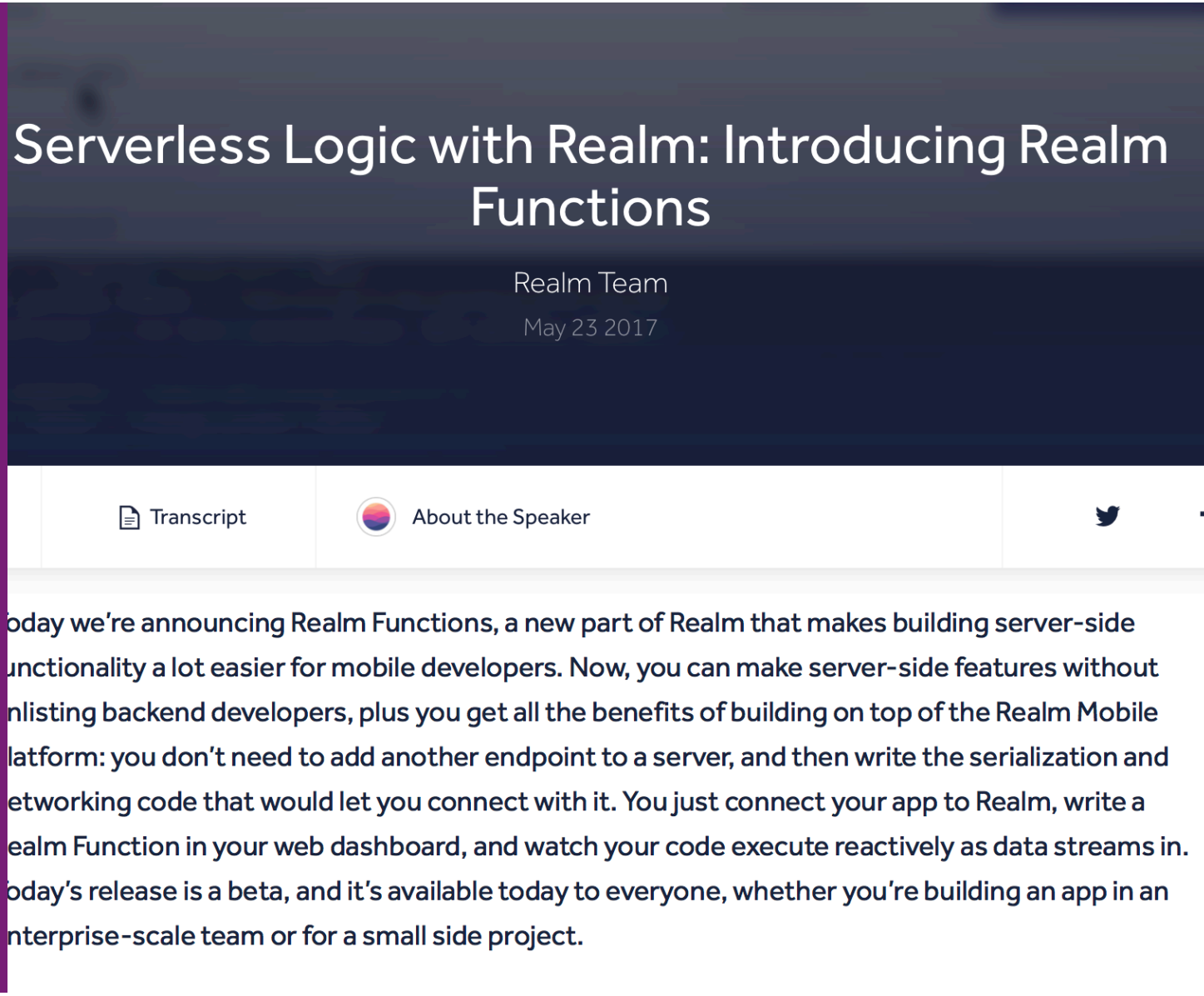
REALM

- Local cache and sync
- Live objects with observer model (like Firebase)
- Subscriptions (like CloudKit)

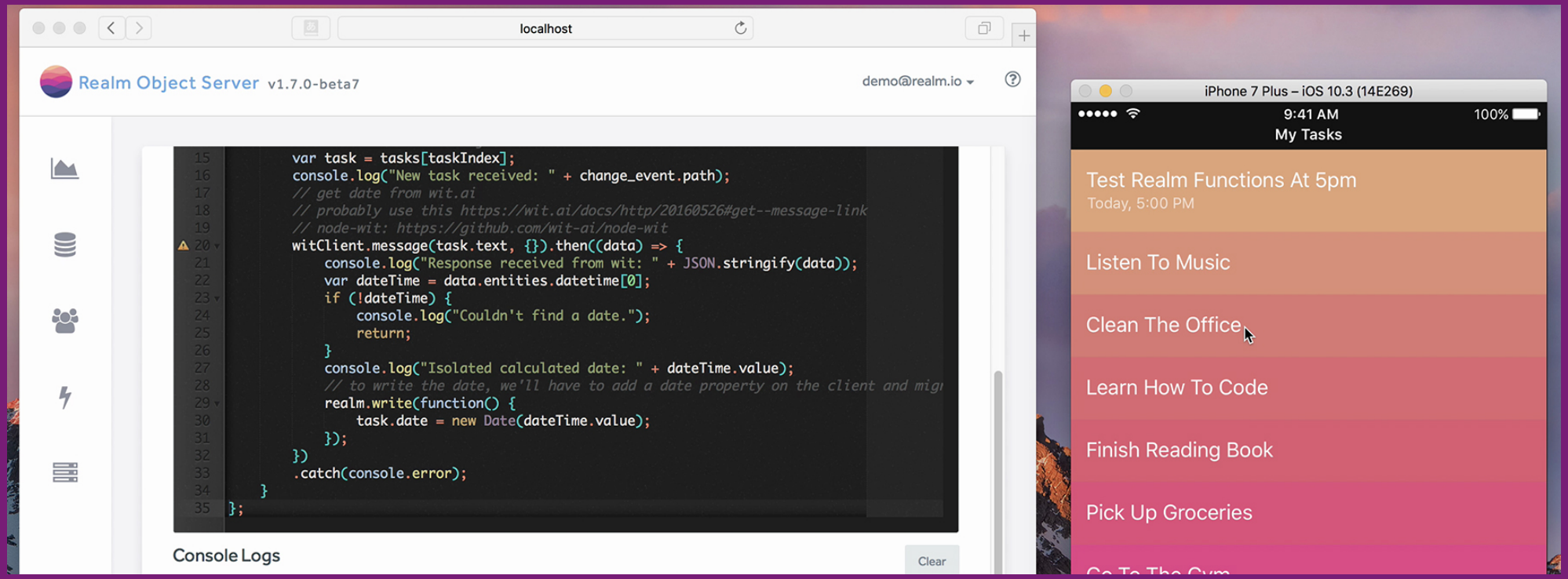


REALM

- But limitation of no server side logic, until....



REALM

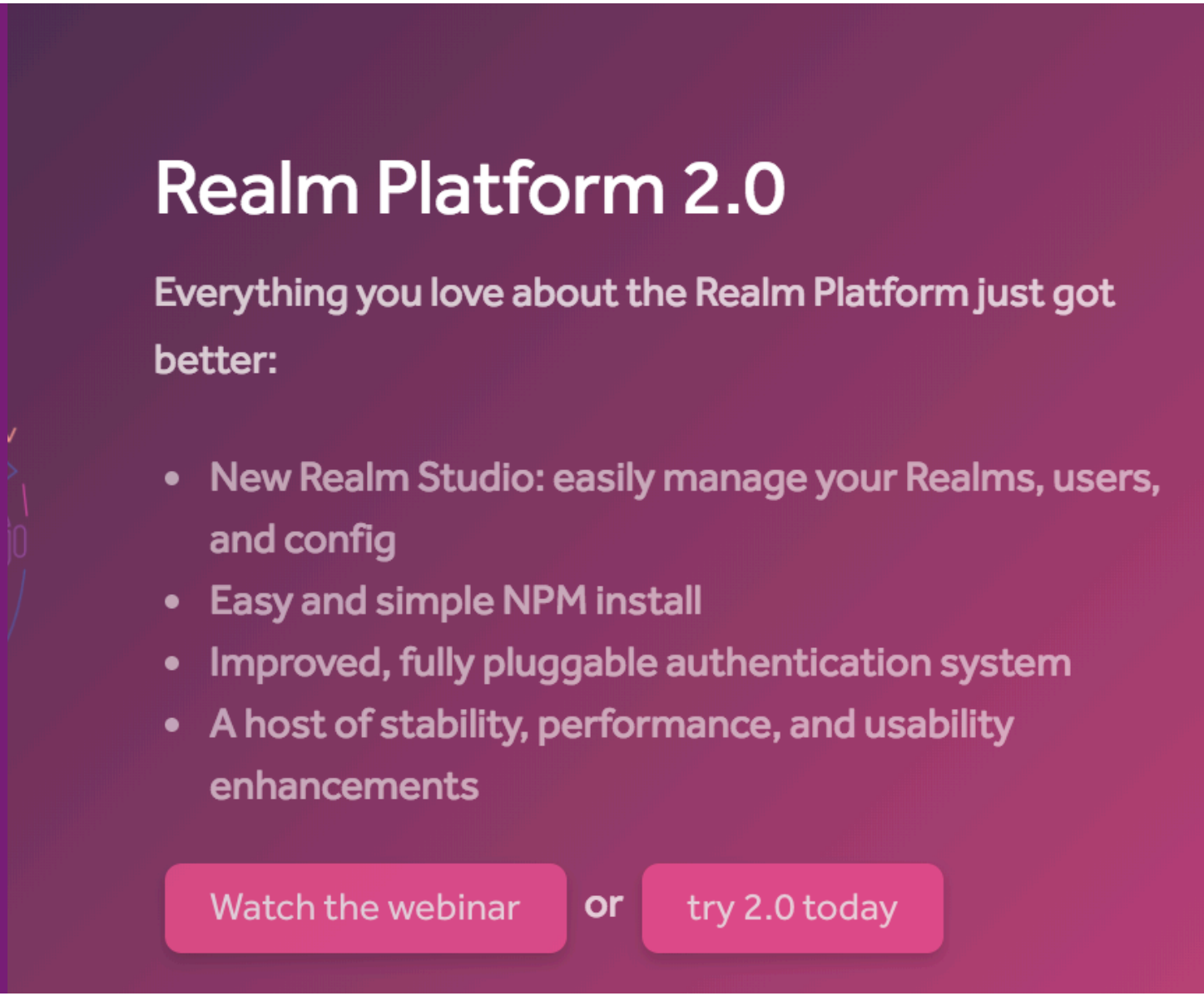


- https://news.realm.io/news/serverless-logic-with-realm-introducing-realm-functions/

REALM PLATFORM

REALM
PLATFORM

- Database
- Platform
- Studio



REALM PLATFORM

- Realm Object Server (ROS)
  - Realtime sync objects (no REST API)
  - Objects are live (no transport layer)
  - No networking code (that you write)

REALM PLATFORM

```
curl -s https://raw.githubusercontent.com/realm/realm-object-server/master/install.sh | bash

ros init my-app

cd my-app/
npm start
```

- Supports Ubuntu, Mac, Windows
  - Public/private clouds
- Run install script on server to download and configure

REALM PLATFORM

- Supports Ubuntu, Mac, Windows
- Run install script on server to download and configure

DOES MORE THAN SYNC

Realm Database

Our fast and reactive database is superior to SQLite-based alternatives as an embedded "live object" database on the device. And when you connect it to Realm Object Server, it becomes a distributed database providing automatic, realtime data synchronization.

Realm Sync

At the heart of Realm Object Server, automatically synchronizing data objects across all devices and the servers in realtime, is Realm Sync. It handles conflict resolution and offline states seamlessly — and your data is safe with TLS/SSL and AES-256 encryption.

Realm Studio

Functioning as your dashboard and your cockpit, Realm Studio gives you control over your data, platform functions, users, and configuration. Featuring an efficient, task-oriented UI, it's built on Electron and it works across every major platform.

Re m Connect

m Object Server, ect converts ex g REST APIs and data so rces to live objects, freeing you to focus on features rather

Event handling

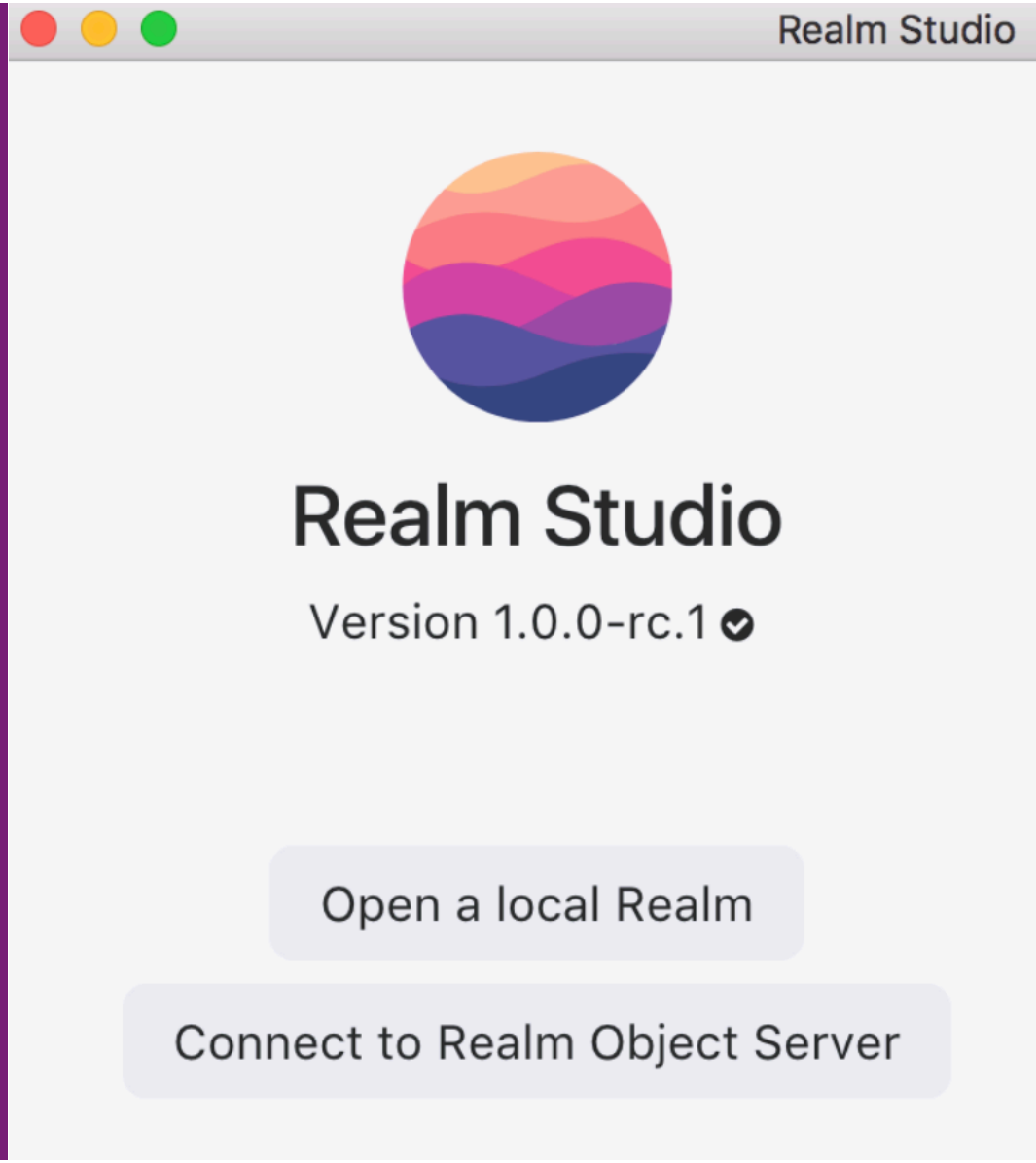
Realm Object Server's event handling functionality means you can easily build server-side features with simple JavaScript. When synced data

Permissions and authentication

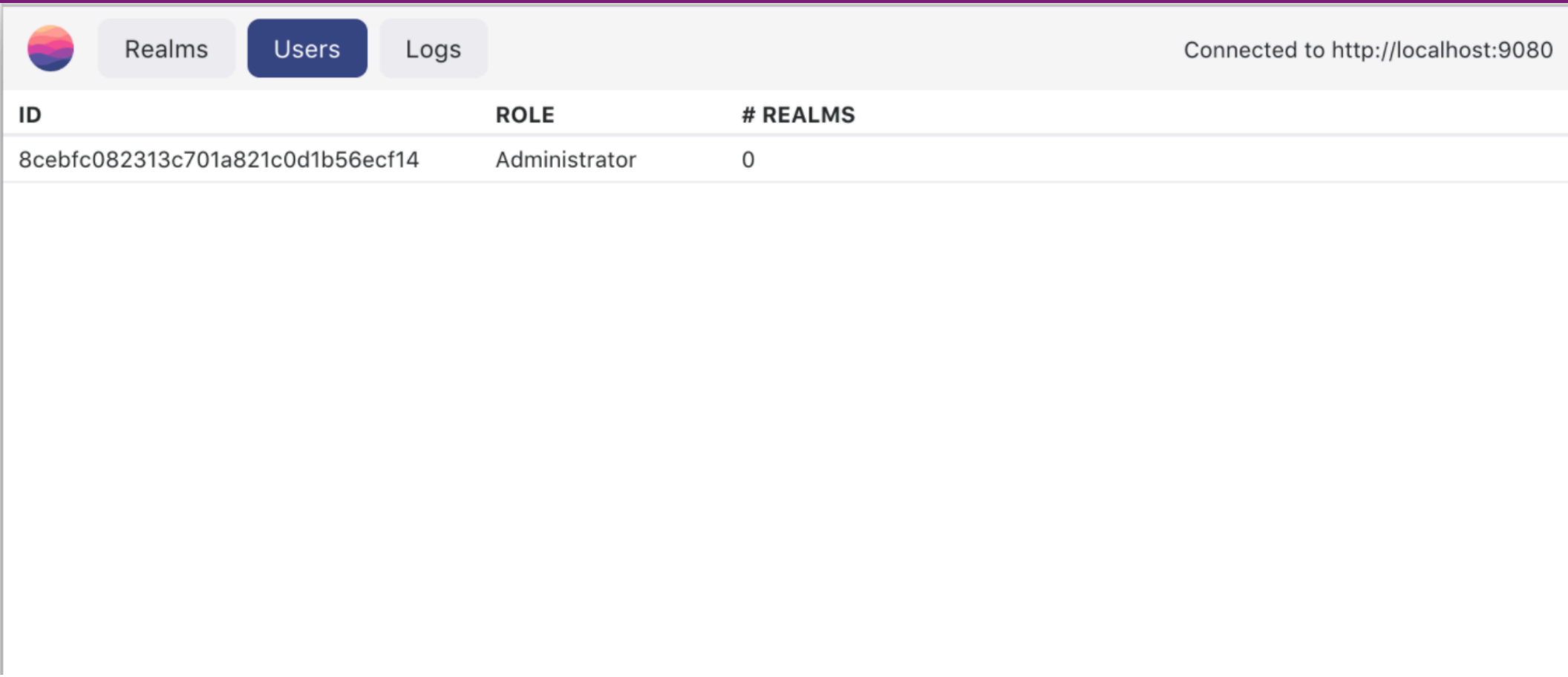
Log in users with our authentication systems, or customize authentication to work with your existing

REALM PLATFORM

- Realm Studio app
  - Manage server data
    - Add, delete, edit
  - View logs

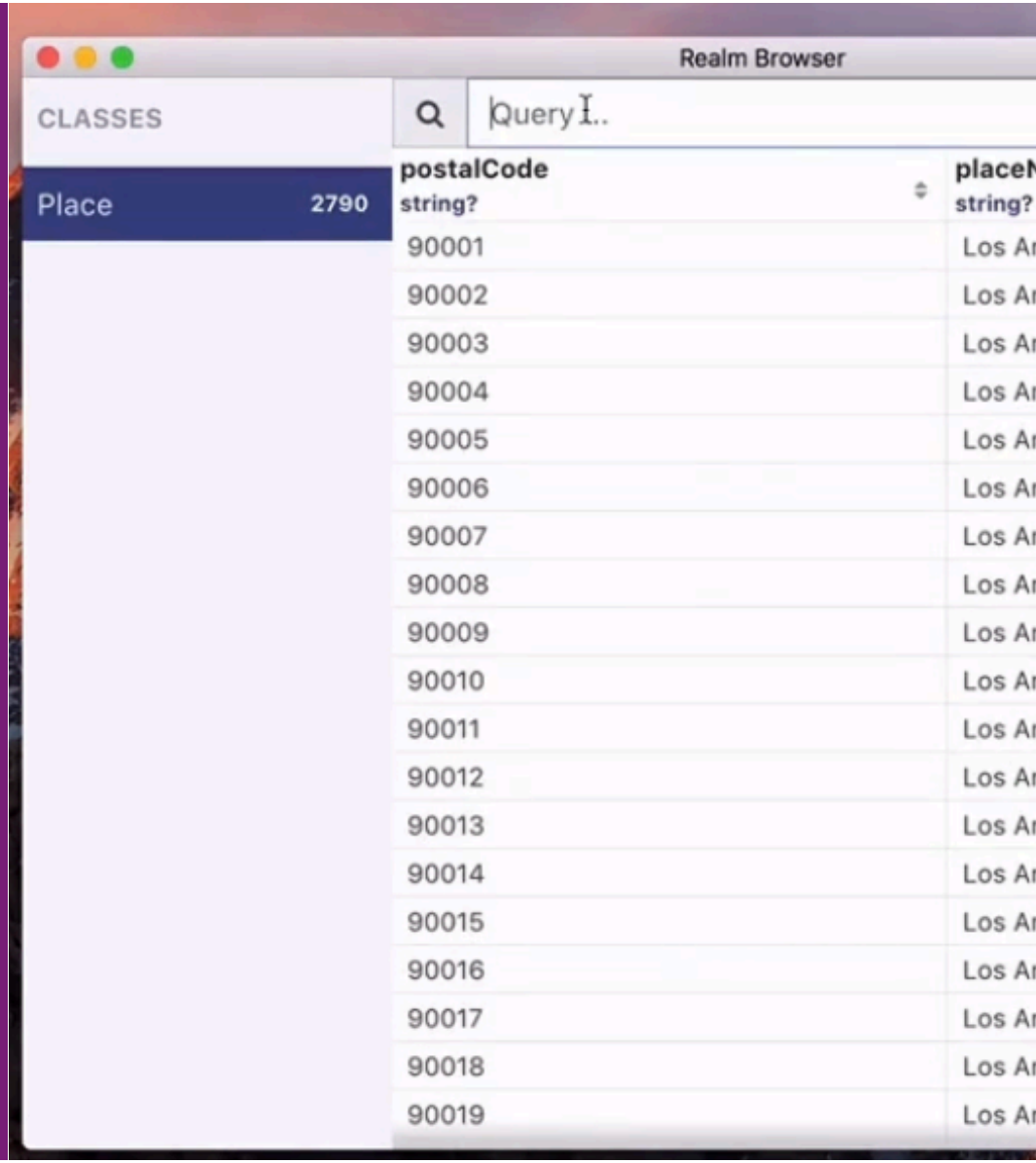


REALM PLATFORM



REALM PLATFORM

- How does the sync engine work?
  - App Engine, explicit calls
  - Firebase 🤷‍♀️
  - CloudKit, subscriptions



REALM PLATFORM

- Realm uses notifications that are sent when it observes a change
  - Realms
  - Collections
  - Objects

```
// Observe Realm Notifications
let token = realm.observe { notification, realm in
    viewController.updateUI()
}

// later
token.invalidate()
```

REALM PLATFORM

- Collections

```
// Observe Results Notifications
notificationToken = results.observe { [weak self] (changes: RealmCollectionChange) in
    guard let tableView = self?.tableView else { return }
    switch changes {
    case .initial:
        // Results are now populated and can be accessed without blocking the UI
        tableView.reloadData()
    case .update(_, let deletions, let insertions, let modifications):
        // Query results have changed, so apply them to the UITableView
        tableView.beginUpdates()
        tableView.insertRows(at: insertions.map({ IndexPath(row: $0, section: 0) }),
                             with: .automatic)
        tableView.deleteRows(at: deletions.map({ IndexPath(row: $0, section: 0)}),
                             with: .automatic)
        tableView.reloadRows(at: modifications.map({ IndexPath(row: $0, section: 0) }),
                             with: .automatic)
        tableView.endUpdates()
    case .error(let error):
        // An error occurred while opening the Realm file on the background worker thread
        fatalError("\(error)")
    }
}
```

REALM PLATFORM

- Objects

```
class StepCounter: Object {
    @objc dynamic var steps = 0
}

let stepCounter = StepCounter()
let realm = try! Realm()
try! realm.write {
    realm.add(stepCounter)
}
var token : NotificationToken?
token = stepCounter.observe { change in
    switch change {
    case .change(let properties):
        for property in properties {
            if property.name == "steps" && property.newValue as! Int > 1000 {
                print("Congratulations, you've exceeded 1000 steps.")
                token = nil
            }
        }
    case .error(let error):
        print("An error occurred: \(error)")
    case .deleted:
        print("The object was deleted.")
    }
}
```

REALM PLATFORM

- Realm takes adventage of Key-Value-Observing (KVO)
- KVO is a mechanism enables objects to register for asynchronous notifications driven by changes in another object's properties

```
import Foundation

//: # Santa object subclass of NSObject #
@objcMembers  // Activate objective-c's introspection capabilit
class Santa: NSObject {

  // The dynamic declaration modifier is required whenever you
  // make use of Objective-C's dynamism
  dynamic var string: String

  override init() {
    string = "🎅"
    super.init()
  }
}

//: # Print original Santa #
let santa = Santa()
print("Original: \(santa.string)")

//: # Set up an observer #
// "\." is shorthand for Santa (in this case); new Key Path in
let observation = santa.observe(\.string) { (santa, change) in
  print("We observed a change in a property.  The new values is
}

//: # Make an observable change #
// We are chaing this...watch what happens
print("Getting read to change....")
santa.string = "santa"
```

REALM PLATFORM

```
import Foundation

//: # Santa object subclass of NSObject #
@objcMembers  // Activate objective-c's introspection capabilities
class Santa: NSObject {

  // The dynamic declaration modifier is required whenever you need to
  // make use of Objective-C's dynamism
  dynamic var string: String

  override init() {
    string = "🎅"
    super.init()
  }
}

//: # Print original Santa #
let santa = Santa()
print("Original: \(santa.string)")
```

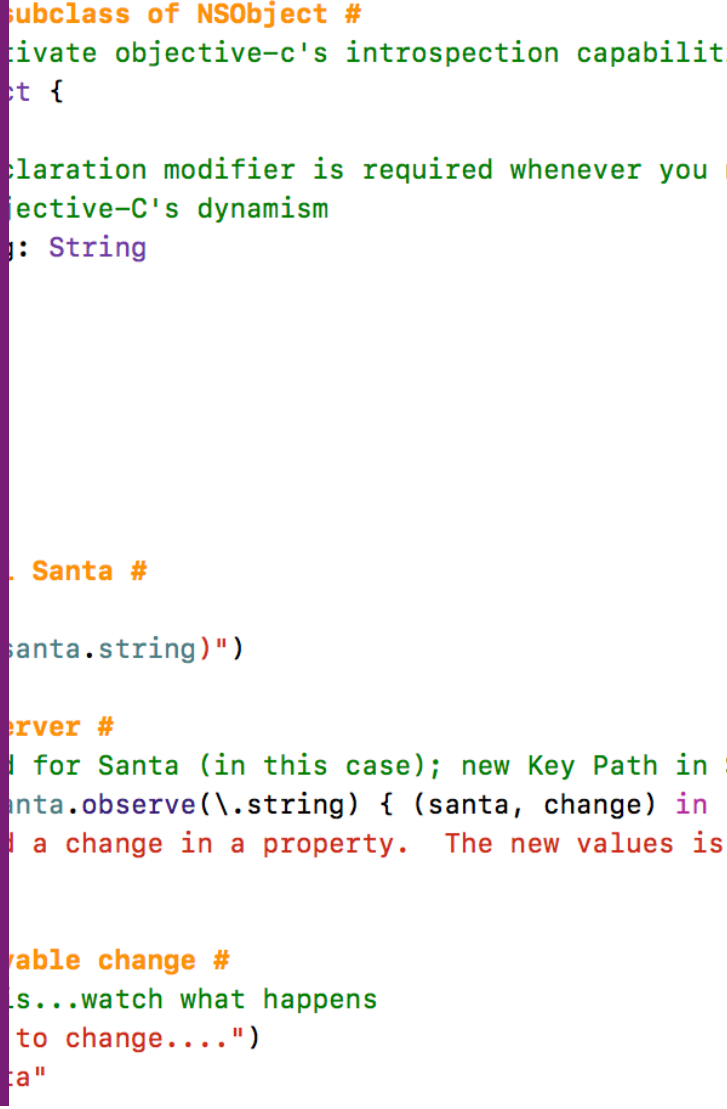
REALM PLATFORM

- When a change happens on device, it triggers update to server
- When a change happens on server, it triggers update on device

```
import Foundation

//: # Santa object subclass of NSObject #
@objcMembers  // Activate objective-c's int
class Santa: NSObject {

  // The dynamic declaration modifier is re
  // make use of Objective-C's dynamism
  dynamic var string: String

  override init() {
    string = "🎅"
    super.init()
  }
}

//: # Print original Santa #
let santa = Santa()
print("Original: \(santa.string)")

//: # Set up an observer #
// "\." is shorthand for Santa (in this cas
let observation = santa.observe(\.string) {
  print("We observed a change in a property
}

//: # Make an observable change #
// We are chaing this...watch what happens
print("Getting read to change....")
santa.string = "santa"
```

REALM PLATFORM

- Similar to CloudKit subscriptions in that changes that match a predictate trigger a notification to be sent to all clients
  - Very limited amount of work setting it up and handling the notifications in CloudKit



REALM PLATFORM

- https://academy.realm.io/posts/learning-path-build-a-realtime-swift-app-with-realm/

Build a Realtime Swift App with Realm

Get hands-on with the Realm Platform, build a realtime Santa-tracking app, and learn ne...

REALM PLATFORM

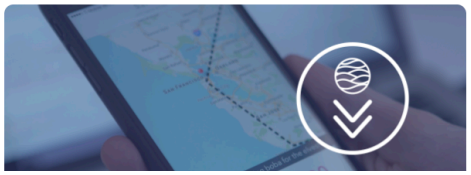
- https://academy.realm.io/posts/learning-path-build-a-realtime-swift-app-with-realm/

Track Santa with Realm: Part 1

Learn how to build a Swift app with the Realm Platform by tracking Santa on his global tour!

Track Santa with Realm: Part 2

Build on your app by completing the data models and making your UI reactive to data changes.

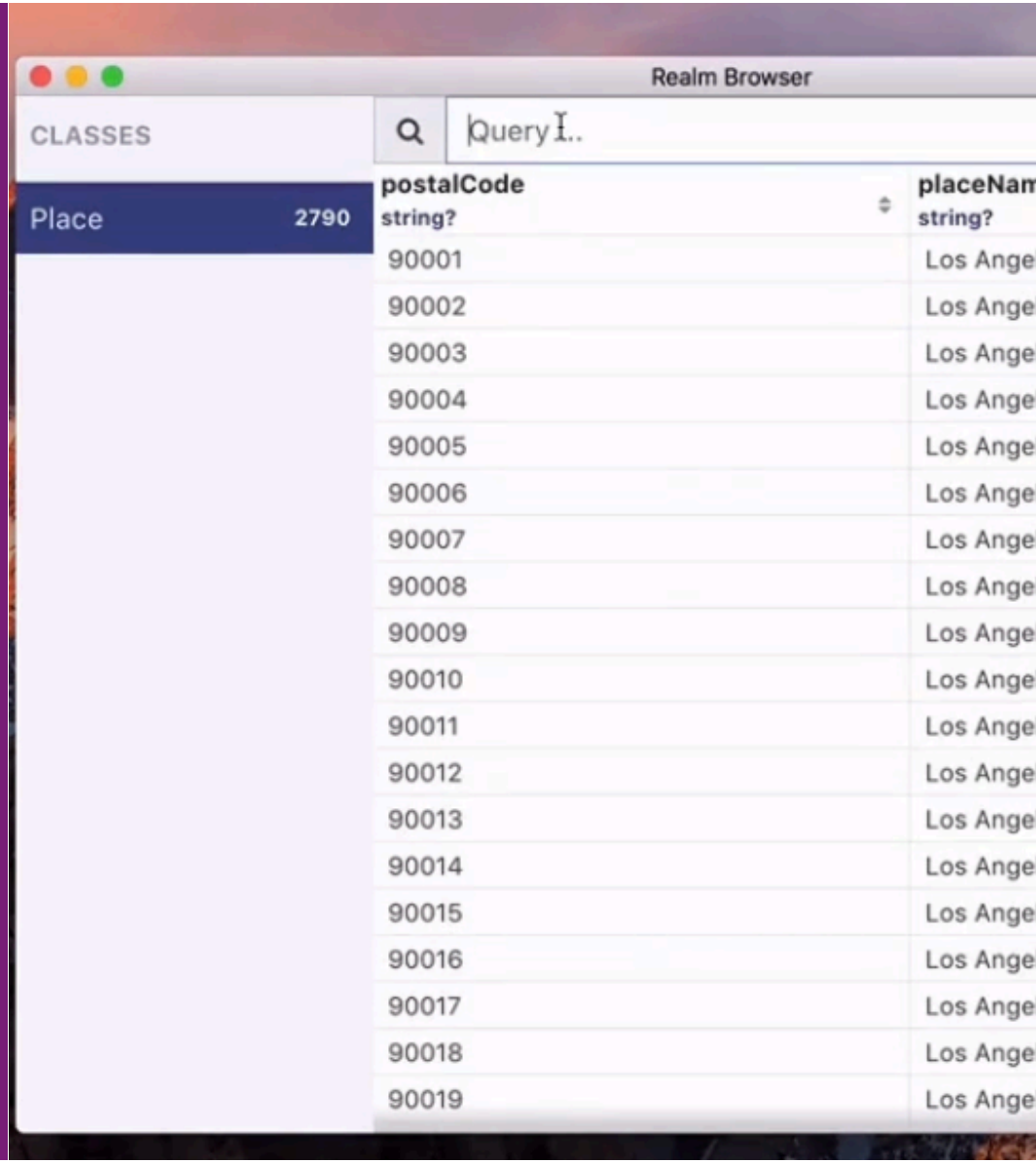


Track Santa with Realm: Part 3

TO REALM OR NOT TO REALM

TO REALM

- Realm checks off all the boxes for a mobile backend
  - Local persistence
  - Sync
  - SDK with networking
  - Authentication



REALM PLATFORM

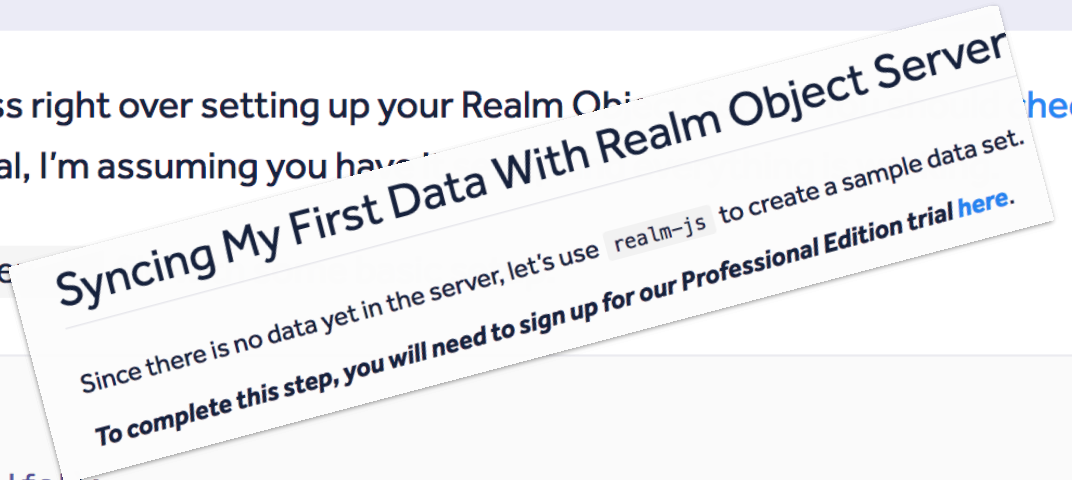
2. Handling the data on the server

This is the cool part. Where the magic happens. Granted, this example isn't very *exciting* magic, but hey, it's magic.

> Server-side data access is only available through the Professional or Enterprise Editions of the Realm Platform. You can read more about those, or sign up for a free trial, on the Pricing page.

So I'm going to gloss right over setting up your Realm Ob… …heck out the… …how to do that. For this tutorial, I'm assuming you hav…

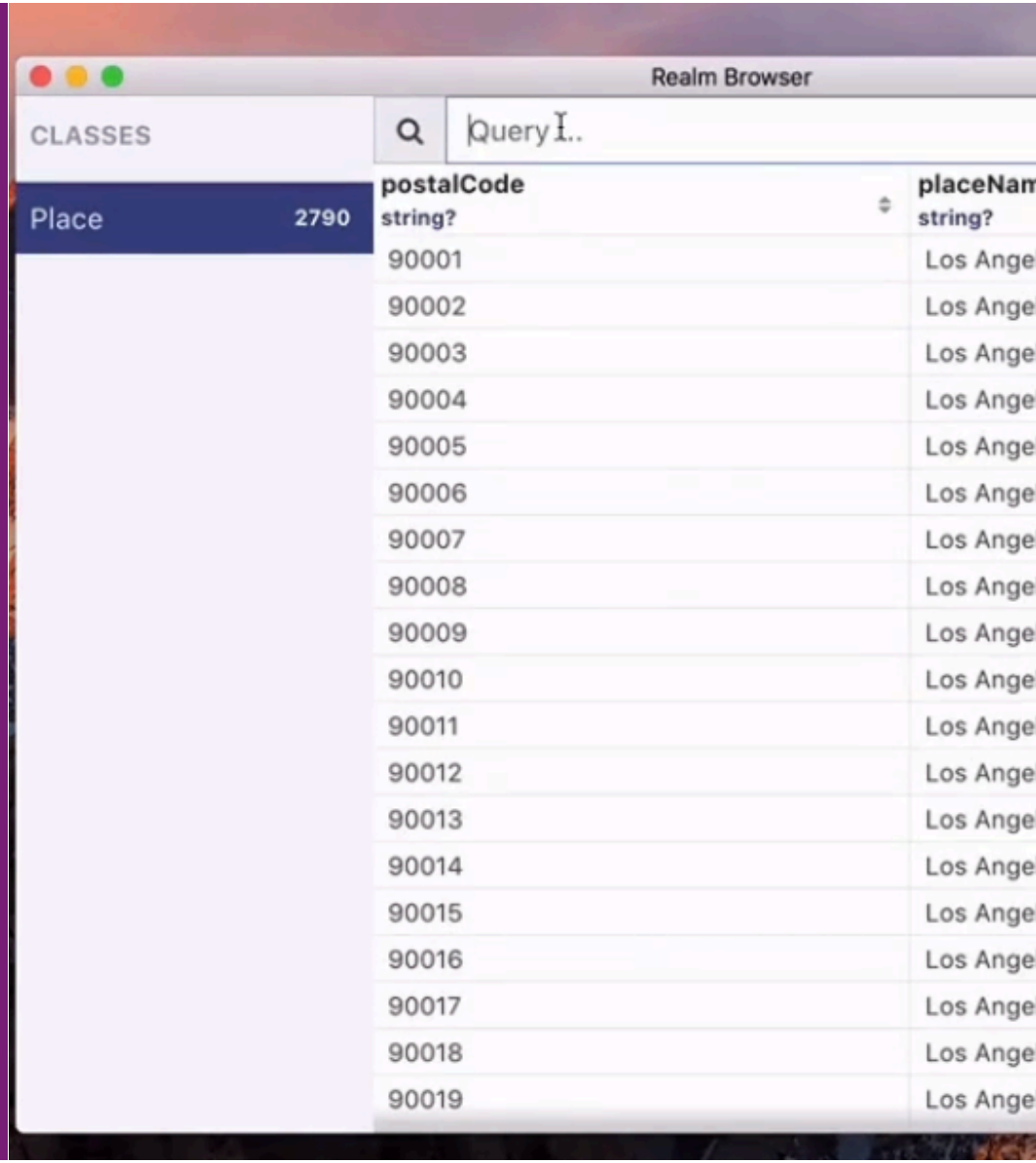
We'll open our `inde…`




```
'use strict';
```

TO REALM

- Professional edition
  - Event handling (functions)
  - Server access
  - Data migration tools
  - Horizontal Scaling?



REALM PLATFORM

Developer Edition

Free

Enable realtime two-way data sync for mobile devices and servers.

No cost to use, even in production for commercial use cases. Read the terms.

Get the download

Professional Edition

Free Trial

Powerful features like simplified integrations, event handling, server-side data access.

14-day free trial; pricing starts at $1,750 per month.

Download free tri

Enterprise Edition

Free Demo

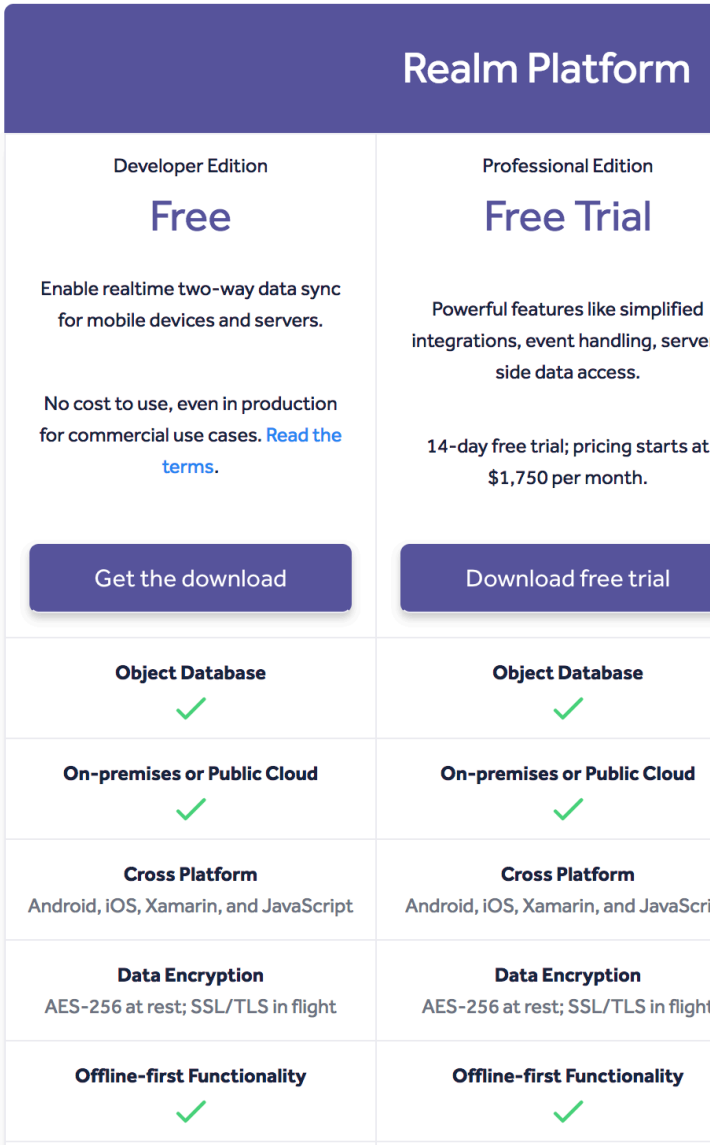
Enterprise scale, availability, and integration with existing infrastructure/third-party APIs.

Pricing scales with use case.

Contact us for d o

TO REALM

- A hybrid of Firebase and CloudKit
  - Less blackbox
  - Requires some "fiddling" on the server
- Privacy (end-to-end control)
- Cross-platform



CLASS IN REVIEW

COURSE DESIGN

COURSE DESIGN

- Today an app is not enough
  - Complexity
  - Interconnected
  - User expectations
  - Insights/analytics
- Provide exposure to technologies to complete mobile applications

This was the goal

COURSE DESIGN

COURSE DESIGN

- Current state of the field
  - Technologies come and go
  - Consolidation/duplication of services
  - DIY mentality
  - Industry proving ground doesn't fit with the narrative

Google App Engine

Windows Azure
The Future Made Familiar

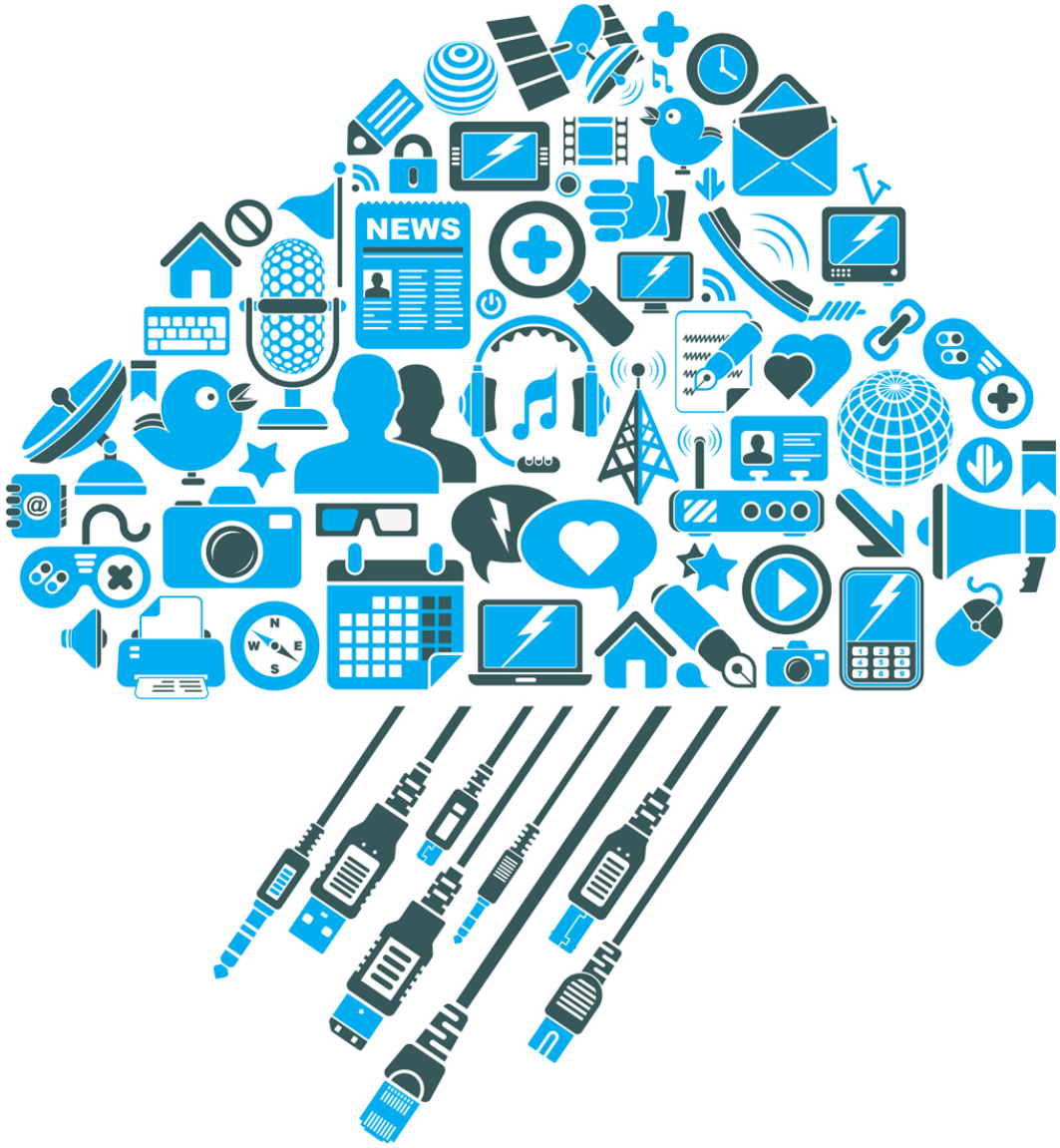
amazon web services™

rackspace®
HOSTING

WHERE WE STARTED

BACKENDS FOR MOBILE APPLICATION

- Backends is a specific utilization of cloud computing



BACKENDS FOR MOBILE APPLICATION

- Platforms
  - Provides on demand services
  - Pay for use and as needed, elastic
  - Scale up and down in capacity and functionalities
  - The hardware and software services are available to general public, enterprises, corporations and businesses markets

BACKENDS FOR MOBILE APPLICATION

Common Characteristics:

- Massive Scale
- Resilient Computing
- Homogeneity
- Geographic Distribution
- Virtualization
- Service Orientation
- Low Cost Software
- Advanced Security

Essential Characteristics:

- On Demand Self-Service
- Broad Network Access
- Rapid Elasticity
- Resource Pooling
- Measured Service

BACKENDS FOR MOBILE APPLICATION

- Cloud service models
  - Software as a service (Saas)
    - Adobe Creative Cloud
  - Platform as a service (PaaS)
    - S3
  - Infrastructure as a service (Iaas)
    - Compute Engine

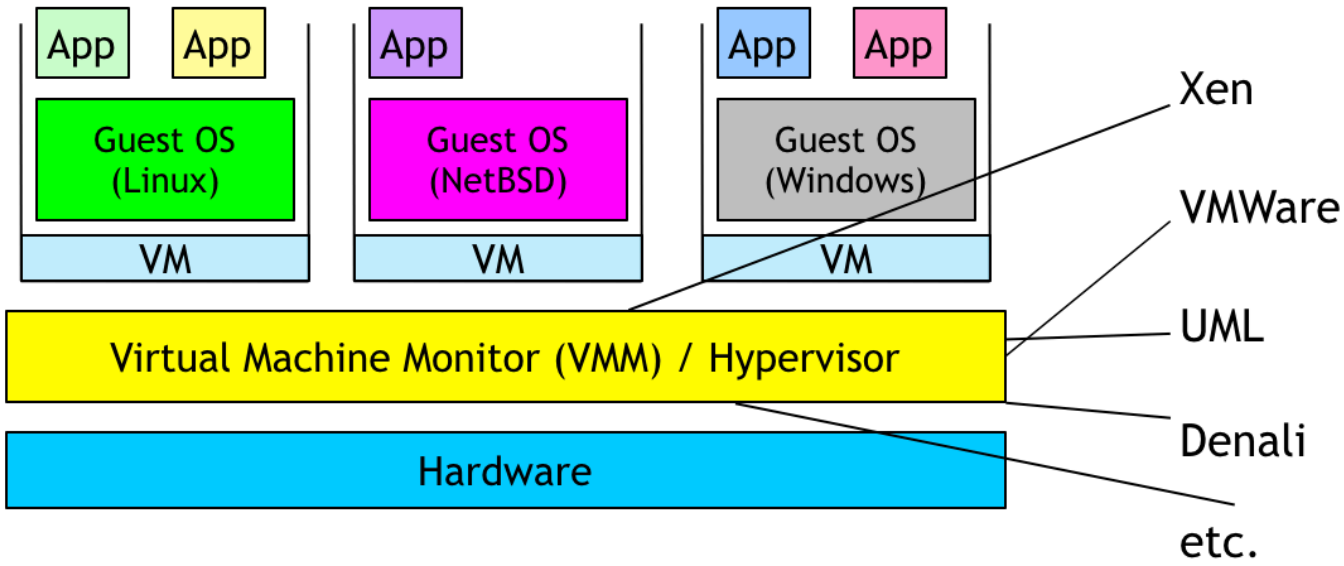
BACKENDS FOR MOBILE APPLICATION

- Different Cloud Computing service layers

	Services	Description
Application Focused	Services	Services - Complete business services such as PayPal, OpenID, OAuth, Google Maps, Alexa
	Application	Application - Cloud based software that eliminates the need for local installation such as Google Apps, Microsoft Online
	Development	Development - Software development platforms used to build custom cloud based applications (PAAS & SAAS) such as SalesForce
Infrastructure Focused	Platform	Platform - Cloud based platforms, typically provided using virtualization, such as Amazon ECC, Sun Grid
	Storage	Storage - Data storage or cloud based NAS such as CTERA, iDisk, CloudNAS
	Hosting	Hosting - Physical data centers such as those run by IBM, HP, NaviSite, etc.

BACKENDS FOR MOBILE APPLICATION

- VM (virtual machine) technology allows multiple virtual machines to run on a single physical machine
- Abstraction of a physical machine



BACKENDS FOR MOBILE APPLICATION

- Advantages
  - Performance, cost, access, collaboration, reliability, bought expertise, security
- Disadvantages
  - Need a network, give up control, monopolization of industry, security

WHERE WE ENDED

BACKENDS FOR MOBILE APPLICATION

- Platforms
  - App Engine
  - Firebase
  - iCloud (CloudKit)
  - Hosted VMs with Swift Server Frameworks
  - Realm

BACKENDS FOR MOBILE APPLICATION

- App Engine
  - The most complete offering
  - Flexibility without (too much) complexity
  - Can do everything..plus and a minus
  - Local development environment
  - All client work is up to developer

BACKENDS FOR MOBILE APPLICATION

- Assignment 2
  - Use app engine to create a photo timeline app
  - Designed CRUD API
  - Basic authentication
  - Use Google Vision API

Assignment 2 Part 2
Mobile Analytics Platform

In this part of the assignment, you will create a mobile analytics platform to collect information about how customers are using your application. The platform should collect and aggregate the following information:

- **User Session.** A user session is defined as the amount of time an application is continuously active. For example, if a user opens the app uses it for 30 seconds and then switches to another app, this would be a single session of length 30 seconds. Later that day, if the user opens the app and uses it for 180 seconds, that would be a second user session of 180 seconds.

- **User Events.** A user even is recorded when the user performs a specific action in the application. For example, you may choose to record button taps, segue to specific view controllers, or the numbers of times that a user reloaded a page. *For this application and simplicity, we will only be concerned with a button tap user event.*

- **Touch Heat Map.** Our app is designed for single handed use and our UI/UX team is considering how the screens size of the "plus" phones is effecting the usability of our app. We want to collect all the touches that a user makes in a given session.

Every day at 10pm, you should send an email to the developer with a summary of the apps usage statistics. The information in the email should include the following:

- **Daily Active Users.** This represents the number of unique users that used your app on a given day.

- **Daily New Users.** The number of new users for a given day.

- **Monthly Active Users.** The number of users for the current month

BACKENDS FOR MOBILE APPLICATION

- Assignment 3
  - Created a mobile analytics backend on App Engine
  - Microservice architecture
  - Store data and server logic
  - Cron job to send email summary

Mobile Analytics Platform

In this part of the assignment, you will create a mobile analytics platform to coll
information about how customers are using your application. The platform sho
collect and aggregate the following information:

- **User Session.** A user session is defined as the amount of time an applicati
continuously active. For example, if a user opens the app uses it for 30 se
and then switches to another app, this would be a single session of length
seconds. Later that day, if the user opens the app and uses it for 180 seco
that would be a second user session of 180 seconds.

- **User Events.** A user even is recorded when the user performs a specific a
the application. For example, you may choose to record button taps, seg
specific view controllers, or the numbers of times that a user reloaded a pa
*For this application and simplicity, we will only be concerned with a buttor
user event.*

- **Touch Heat Map.** Our app is designed for single handed use and our UI/
team is considering how the screens size of the "plus" phones is effecting
usability of our app. We want to collect all the touches that a user makes
given session.

Every day at 10pm, you should send an email to the developer with a summary
apps usage statistics. The information in the email should include the following

- **Daily Active Users.** This represents the number of unique users that used
app on a given day.

- **Daily New Users.** The number of new users for a given day

BACKENDS FOR MOBILE APPLICATION

- Firebase
  - Ease of use, broad spectrum of offerings
  - Extensible with GCP
  - Mobile first libraries
  - Unique mobile-first offerings (invites, authentication, etc.)
  - Structing data is not-intuitive

BACKENDS FOR MOBILE APPLICATION

- Assignment 4
  - Chat app using Firebase
  - Dynamic links with invites
  - Structuring a database with
  - Cloud functions
  - Integrate with mobile app

Assignment
By Invitation O
An Exclusive Cha

In this assignment, you will create a group chat applicat
many features available through the Firebase platform.
common functionality of group chat applications to allo
each other.

Overall Application Behavior

All users will have to be authenticated with Google auth
After signing in, users should be presented with a list of
Navigating to one of these groups should take them to
application. Users should be allowed to post a new cha

Chat groups are **invite-only** and the only way to join on
an existing member. Users of the application will be all
hence becoming the first member, and then be allowed

BACKENDS FOR MOBILE APPLICATION

- CloudKit
  - Integration with iOS
  - No local storage
  - Finicky 🤷‍♀️
  - Subscriptions
  - Server to server API

BACKENDS FOR MOBILE APPLICATION

- Assignment 4
  - Joke of the day using CloudKit
  - Server to server logic offloaded to Google App Engine (👍)
  - Notification tool

Joke of the Day 😂

In this part of the assignment, you will create a crowd sourced "J
application. The application will be backed by CloudKit with ad
provided by a microservice running on Google App Engine.

Overall Application Behavior

Users will be able to submit a joke through the iOS application.
comprised of both the text of the joke and audio of the joke bei
able to view and listen to all the jokes and rate them. All users s
subscription set up so that they received a notification when a ne
public database.

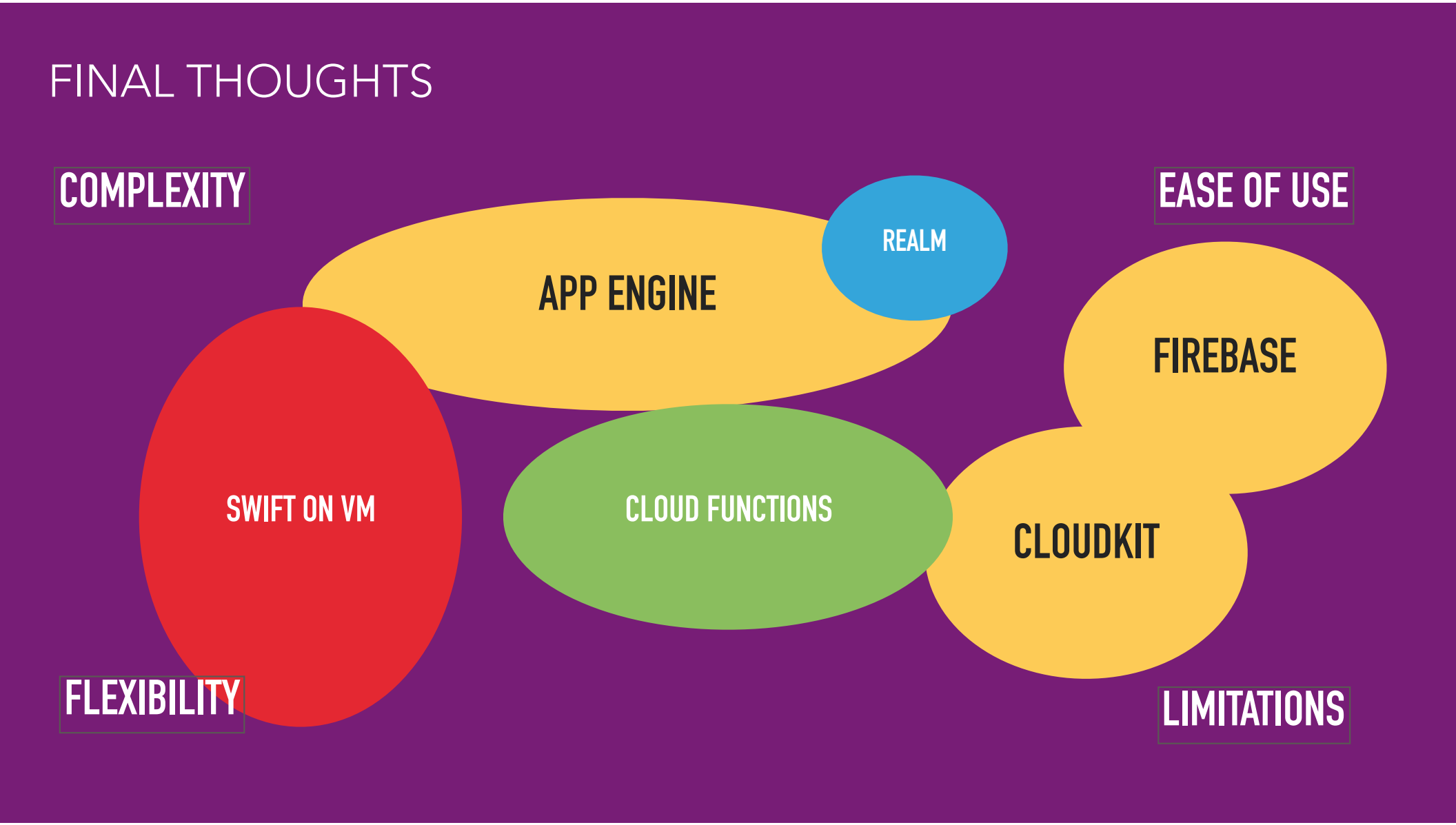
Each morning, users will receive a notification informing them of
or the highest rated joke for the previous day. This functionality
cron job service on Google App Engine that communicates with
server communication. At a set time, the service should request
the previous day and sort them by their rating. The top rated on
the Day."

BACKENDS FOR MOBILE APPLICATION

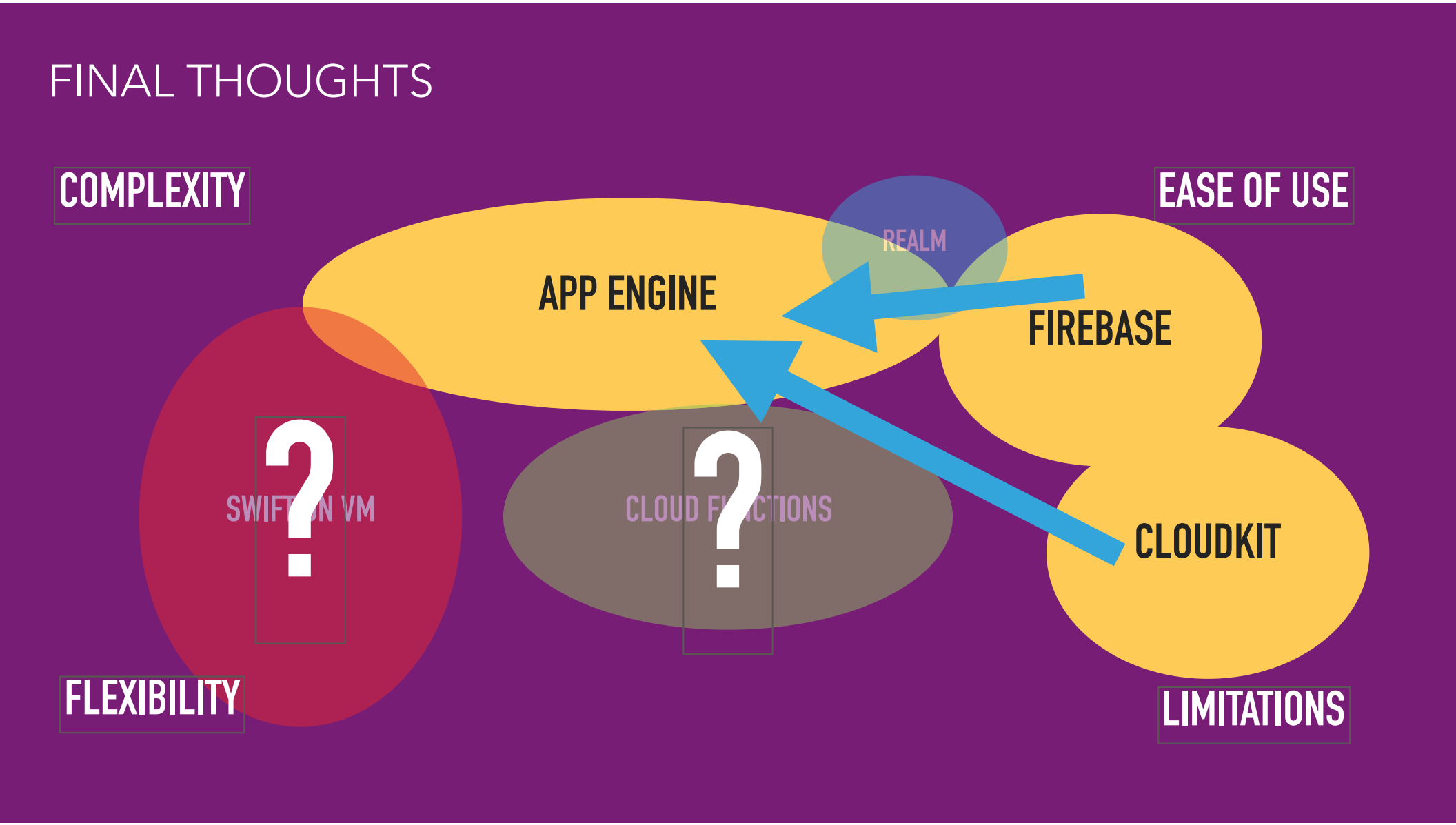
- VM with Swift on the Server
  - Vapor or Perfect
  - Integration with cloud based PostgreSQL server
  - Deploy Heroku, AWS, Google Cloud

FINAL THOUGHTS

FINAL THOUGHTS



FINAL THOUGHTS



FINAL THOUGHTS

- Is there a clear winner? Nope
- You can make them all do the same thing
- Decision points
  - Client platform
  - Service familiarity
  - Integrations

FINAL THOUGHTS

- Final decision is probably most determined by the mobile app functionality
  - Keep as much work on the client device
  - Backend fills the gap in functionality

THANK YOU

THANK YOU

- Feedback on course, content and design
  - Now, later, much later
- Please stay in touch

THE UNIVERSITY OF CHICAGO

MPCS 51033 • AUTUMN 2017 • SESSION 10

BACKENDS FOR MOBILE APPLICATIONS