Deploy your microservice using
AWS S3,
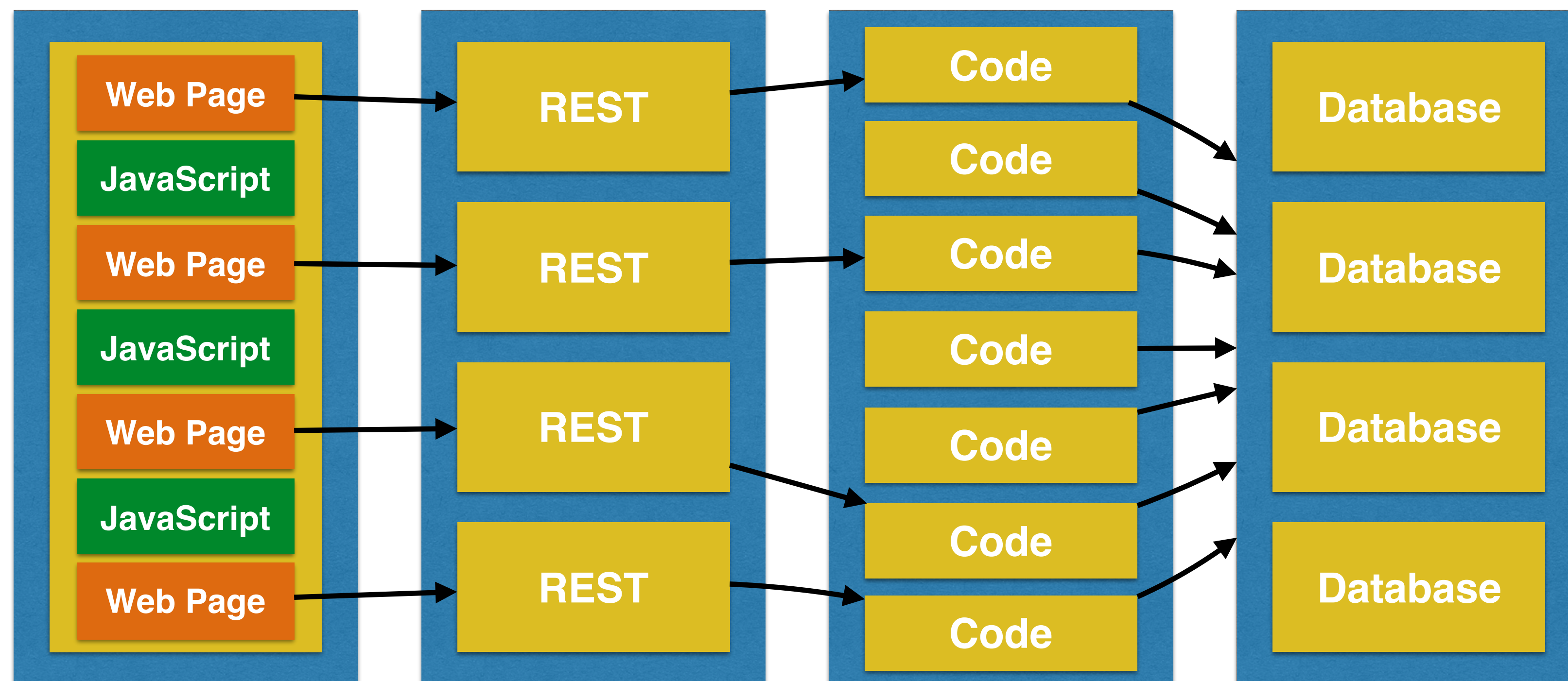# AWS API Gateway,
AWS Lambda,
and
Couchbase

Arun Gupta, @arungupta

Couchbase

*the microservice architectural style is an approach to developing a single application as a **suite of small services**, **each running in its own process** and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and **independently deployable** by fully automated deployment machinery. There is a **bare minimum of centralized management** of these services, which may be written in **different programming languages** and use **different data storage technologies***

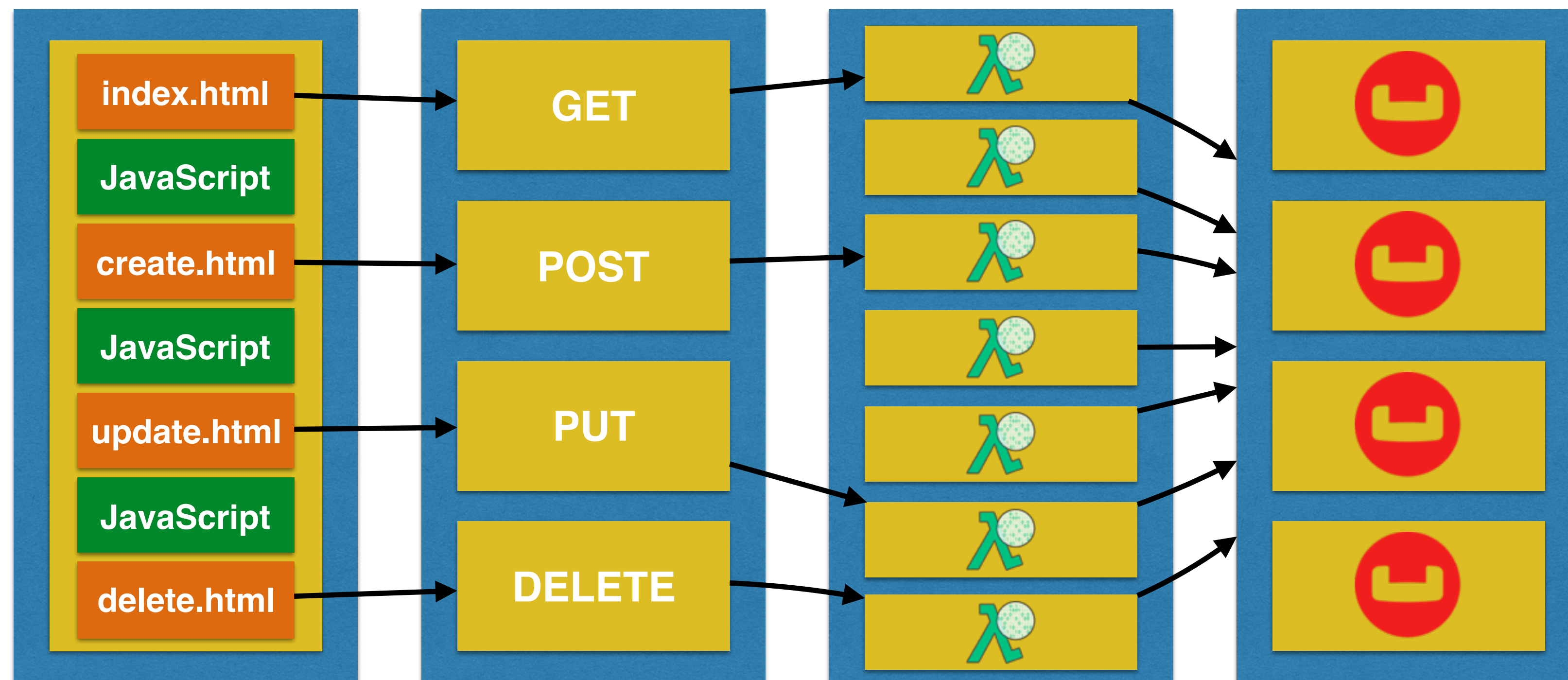https://martinfowler.com/articles/microservices.html

Serverless Computing

Typical Challenges with Server Computing

- What size servers for budget/performance?

- Scale servers up/down?

- What O/S?

- O/S settings?

- Patching?

- Control access to servers?

- Deploy new code to server?

	Virtual Machines	Containers	Serverless
Unit of Scale	Machine	Application	Function
Abstraction	Hardware	Operating System	Language Runtime
Packaging	AMI	Container File	Code
Configure	Machine, storage, networking, O/S	Run Servers, configure applications, scaling	Run code when needed
Execution	Multi-threaded, multi-task	Multi-threaded, single task	Single threaded, single task
Runtime	Hours to months	Minutes to days	Microseconds to seconds
Unit of cost	Per VM per hour	Per VM per hour	Per memory/second per request
Amazon	EC2	Docker, Kubernetes, ECS	AWS Lambda

adrian cockcroft
@adrianco

If your PaaS can efficiently start instances in 20ms that run for half a second, then call it serverless.

> **Julz Friedman** @doctor_julz
> if you think serverless is different than PaaS then either you or I have misunderstood what "serverless" or "PaaS" means

RETWEETS
158

LIKES
207

6:43 AM - 28 May 2016

↩ 10   ♺ 158   ♥ 207

What is Serverless?

Fully Managed

- No provisioning
- Zero administration
- High availability
- Never pay for idle

Developer Productivity

- Focus on code that matters
- Innovate rapidly
- Reduce time to market

Continuous Scaling

- Automatically
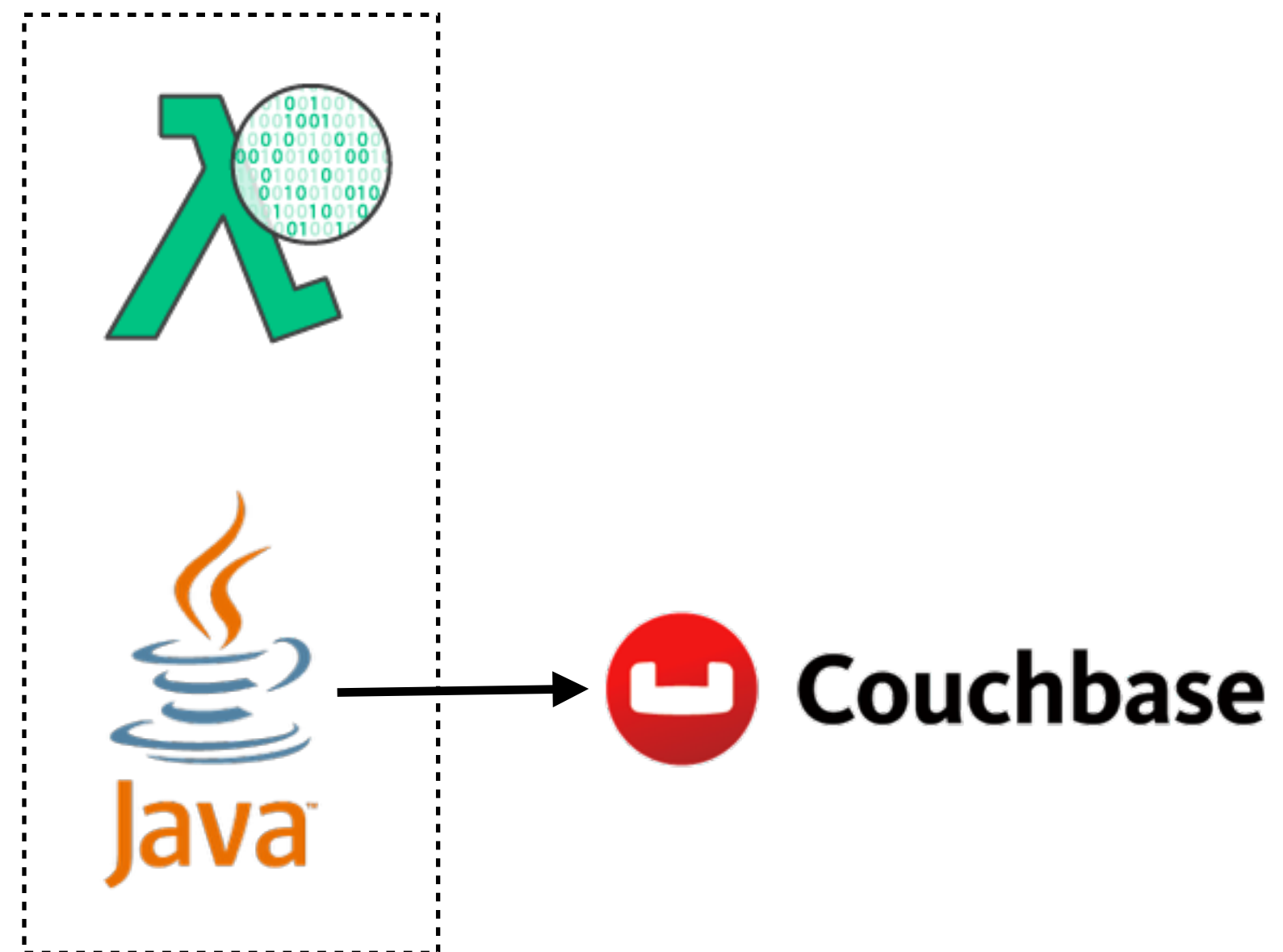- Scale up and down

Key components of AWS Lambda

- A Lambda function

- An event source

- AWS Lambda service

AWS Lambda

- 1M free requests per month

- $0.20 per million requests thereafter

- 400k GB-seconds of compute time per month

Hello Serverless

AWS API Gateway

- Create, publish, maintain, monitor and secure APIs at scale

- API version management: Iterate, test and release new versions

- Easy monitor: Integrated with CloudWatch

- Authorization and access control: integrated with AWS IAM and AWS Cognito

- Low cost and efficient: Only pay for the calls made to APIs and data transfer out

- Traffic management: Set throttle rules

AWS Serverless Application Model

- Standard application model (SAM) for serverless applications
- Extends CloudFormation
- Adds support for API Gateway, AWS Lambda, DynamoDB

AWS S3 Basics

- **Simple**: Console, REST API and AWS SDKs
- **Durable**: 99.999999999%
- **Scalable**: Gigabytes -> Exabytes
  - Store/retrieve data, any time, anywhere
- **Access control**: type of access (e.g. `READ` and `WRITE`)
- **Authentication**: verify identity of the user

AWS S3 Concepts
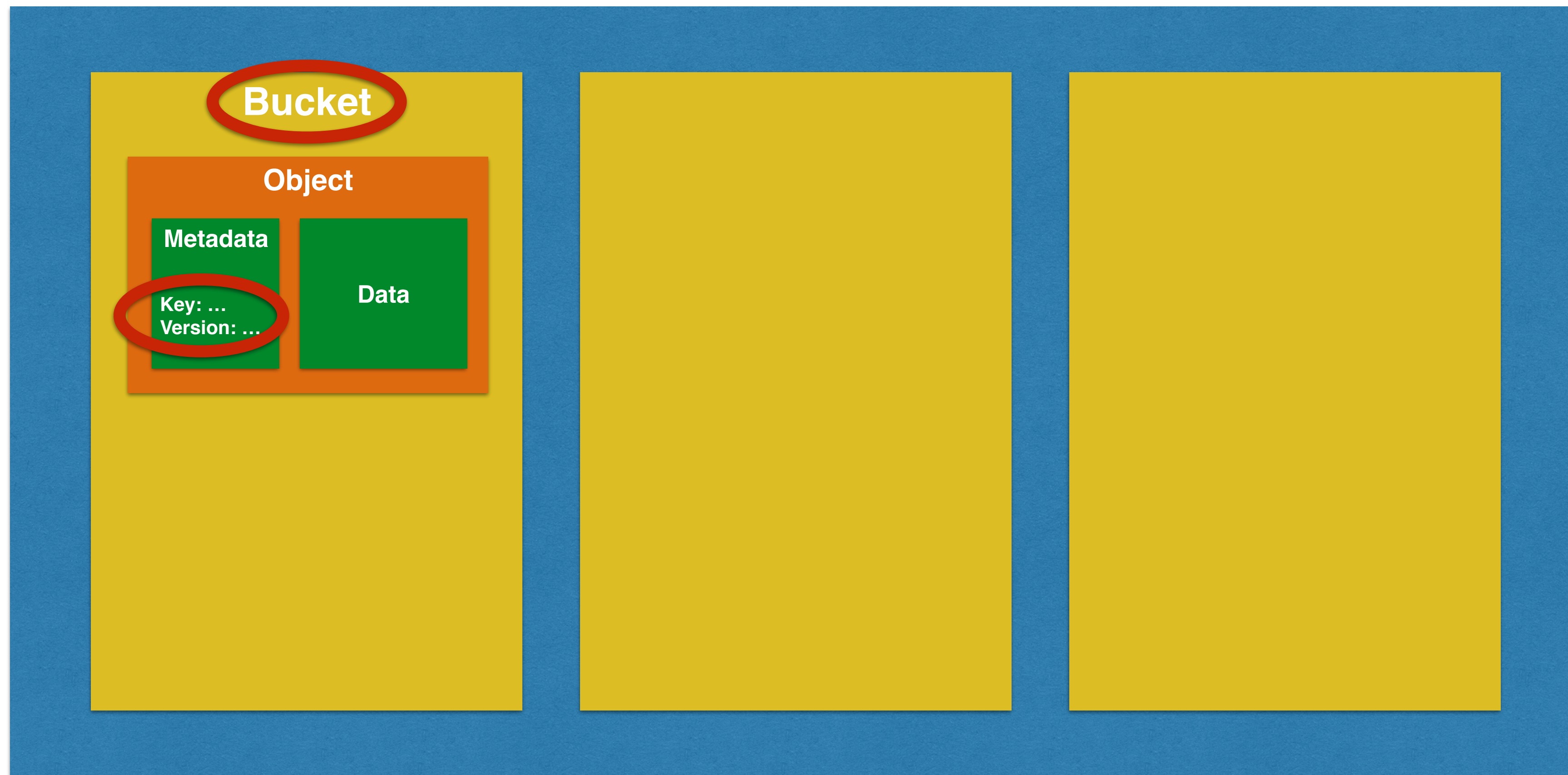
AWS S3 Storage Types

- **Standard**
  - Active data
  - Low latency
  - High throughput
- **Standard Infrequent Access**
  - Infrequent data
  - Same latency/throughput
  - Long term storage, backup, data store for DR
- **Glacier**
  - Archive data
  - Extremely low cost
  - Long term archiving

AWS S3 Storage Pricing

- AWS Free Usage Tier
  - 5GB of standard storage
  - 20k GET, 2k PUT, 15GB transfer

Region: US West (Northern California) ⇕

	Standard Storage	Standard - Infrequent Access Storage †	Glacier Storage
First 50 TB / month	$0.026 per GB	$0.019 per GB	$0.005 per GB
Next 450 TB / month	$0.025 per GB	$0.019 per GB	$0.005 per GB
Over 500 TB / month	$0.024 per GB	$0.019 per GB	$0.005 per GB

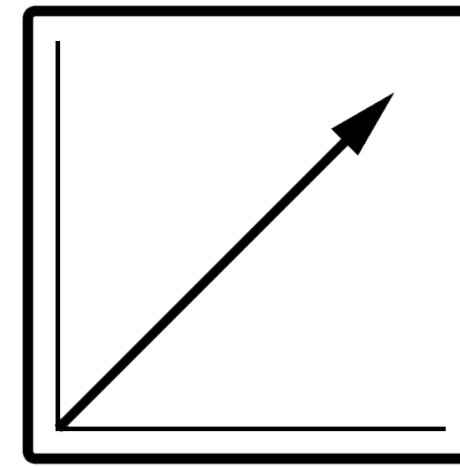https://aws.amazon.com/s3/pricing/
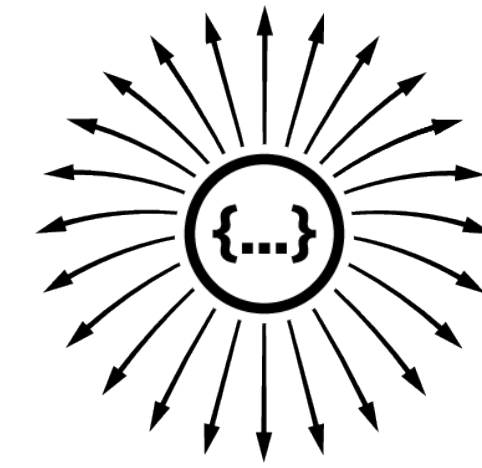
Hosting Static Websites on S3

Couchbase

Responsive
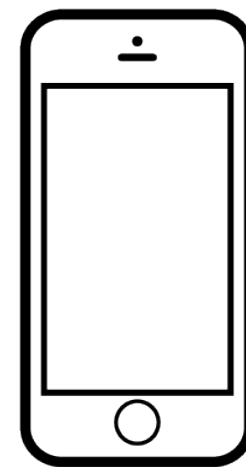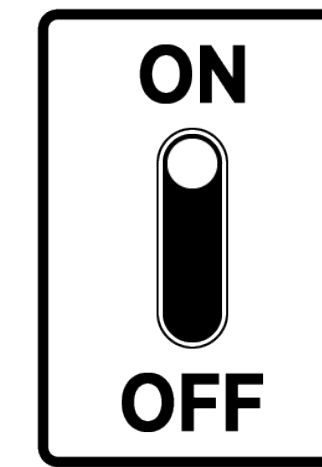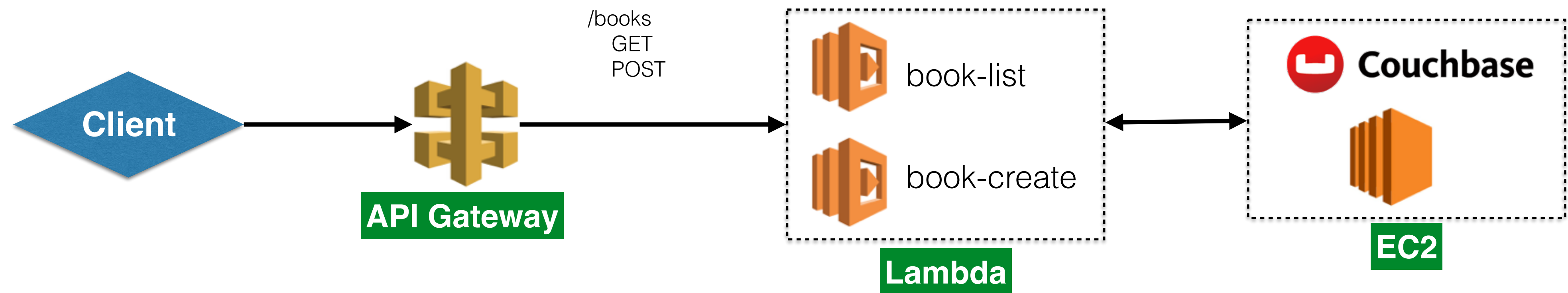
Scalable

Data Agility

Mobility

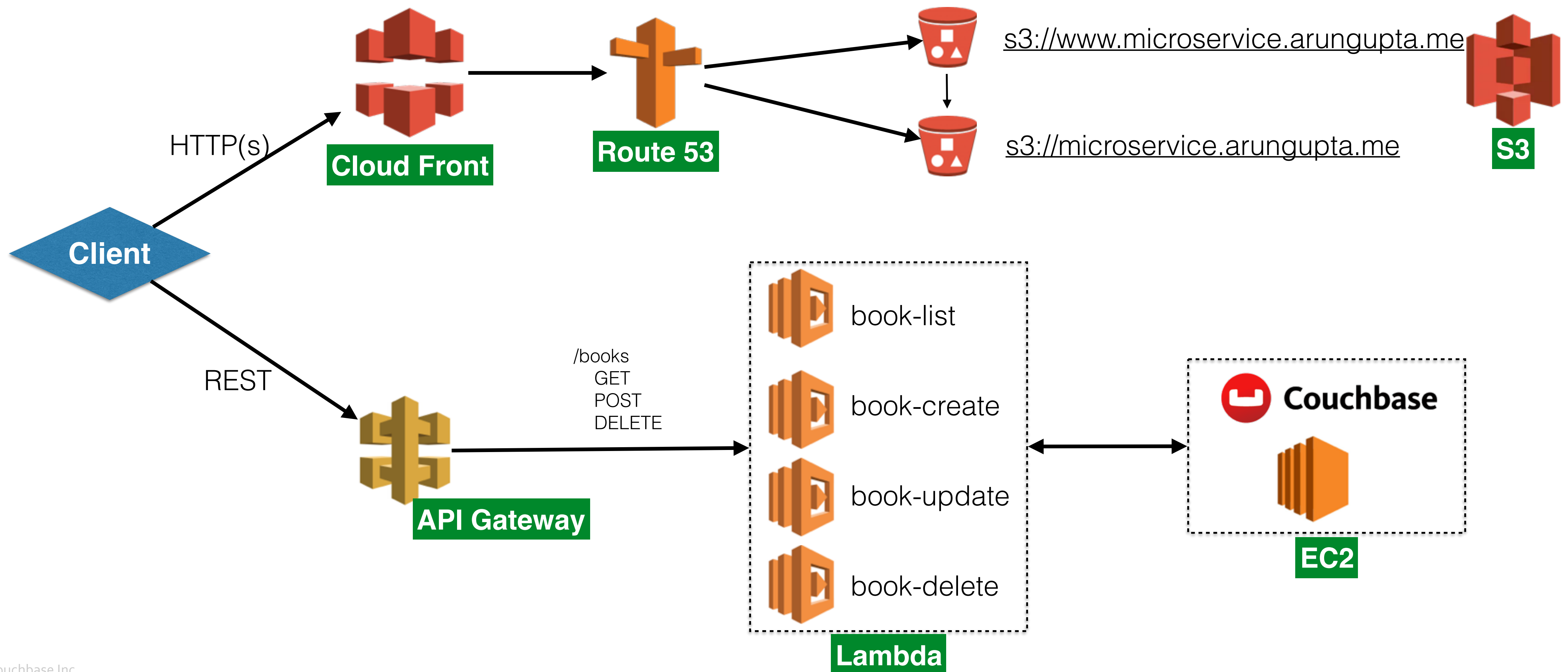Geo-distributed

Easy to Manage

ON
OFF

Typical development workflow

- Write and Deploy a Lambda function

- Create and deploy a REST API with API Gateway

- Connect the API to the Lambda Function

- Invoke the API

- Test, debug and repeat!

Serverless Microservice

Client → API Gateway → /books GET POST → Lambda (book-list, book-create) ↔ EC2 (Couchbase)

Serverless Application

Client

HTTP(s) → **Cloud Front** → **Route 53** →
s3://www.microservice.arungupta.me
s3://microservice.arungupta.me

S3

REST → **API Gateway**

/books
GET
POST
DELETE

Lambda
- book-list
- book-create
- book-update
- book-delete

↔ Couchbase **EC2**

Serverless Application Model

https://github.com/arun-gupta/serverless/blob/master/aws/microservice/template.yml

Twitter

3 hrs

Lambda

EC2