
A:STACK LABS
aws
AWStack #1 Serverless

Win an echo dot

Write a tweet containing :

- @stack_labs
- @AWSFrance
- @goserverless

And **try to Win** An Echo Dot !

(random draw at the end of the day)

Ho !

Fond of development for a long time and recently keen on the AWS Platform

skywise.
Platform Consulting Team

Arthur SUDRE

A:STACK LABS

@asudre

@ASudre

Hey !

Open Source Guru, #hiking #cloud #bigdata #leadership addict, #drone #iot fan, λ\: #stacker translate your vision into powerful architecture solutions

λ\ Chabane REFES

Cloud Guru

λ\: STACK LABS

@ChabaneRefes

@Chabane

About me !

Passionate of backend architecture & development, devops, cloud, more recently interested in serverless architectures

Laurent Noireterre

A:STACK LABS

@LauNoirt

@laurentnoireterre

Introduction

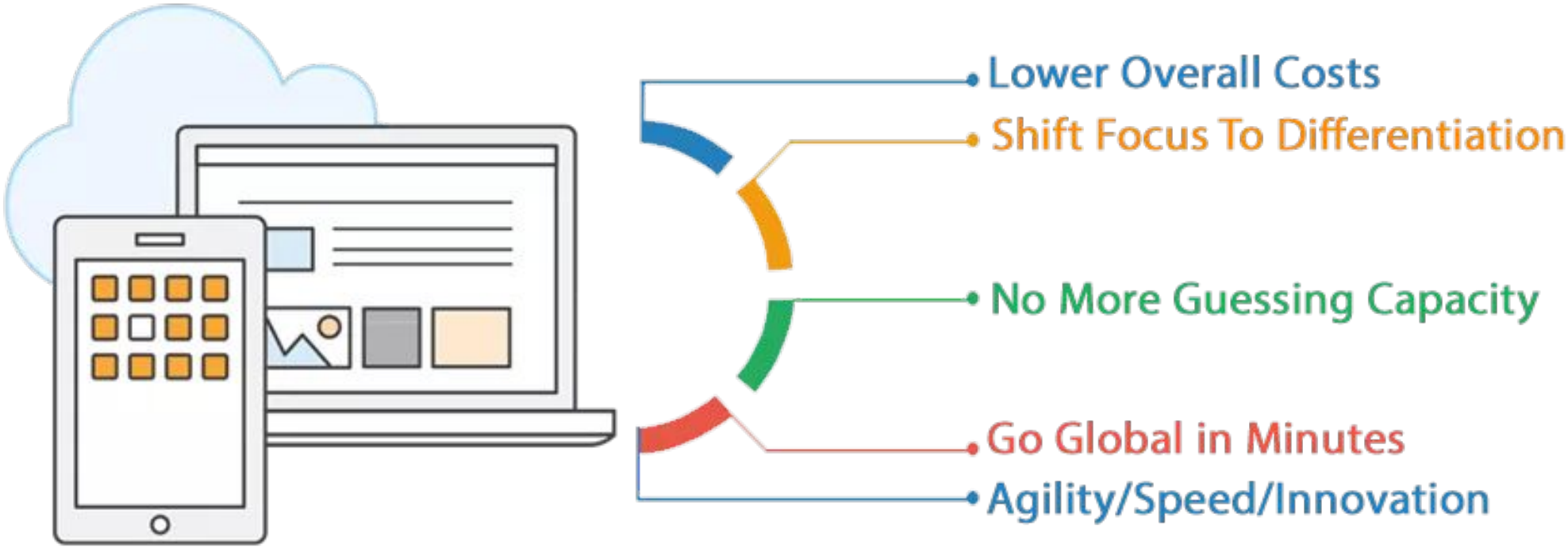
Cloud & Serverless Computing

What is cloud computing ?

*Cloud computing is the **on-demand** delivery of **compute power**, database **storage**, applications, and other **IT resources** through a cloud services platform **via the internet** with **pay-as-you-go** pricing.*

aws

Benefits of Cloud Computing



Source: AWS Documentation

On premise	IAAS	CAAS	PAAS	FAAS	SAAS
Functions	Functions	Functions	Functions	Functions	Functions
Applications	Applications	Applications	Applications	Applications	Applications
Runtime	Runtime	Runtime	Runtime	Runtime	Runtime
Containers	Containers	Containers	Containers	Containers	Containers
Operating System	Operating System	Operating System	Operating System	Operating System	Operating System
Virtualization	Virtualization	Virtualization	Virtualization	Virtualization	Virtualization
Hardware	Hardware	Hardware	Hardware	Hardware	Hardware

EC2

ECS

Beanstalk

Lambda

Platform managed

Customer managed

Serverless means...

No servers to provision or manage

Scales with usage

Never pay for idle

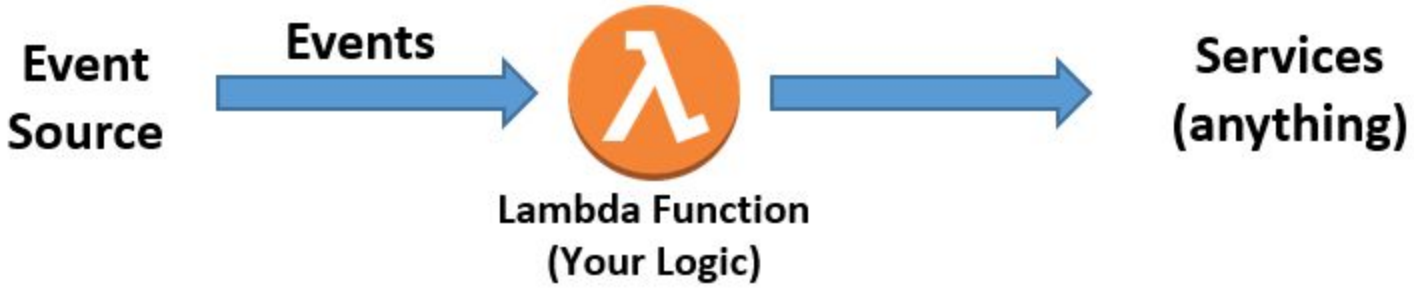
Availability and fault tolerance built in

Source: AWS re:Invent 2017

Full Serverless Applications

Compute	 Lambda
APIs	 API Gateway
Storage	 S3
Databases	 DynamoDB
Messaging	 SNS  SQS
Orchestration	 Step Functions  CloudWatch Events
Authentication	 Cognito

Event driven architecture



Stateless / Decoupling / Concise

Serverless Tooling on AWS

DEV

Cloud9

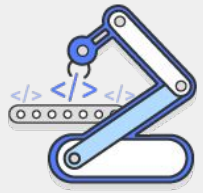
SAM Local

Serverless

CI/CD

CodeStar

Code Pipeline



Code Build

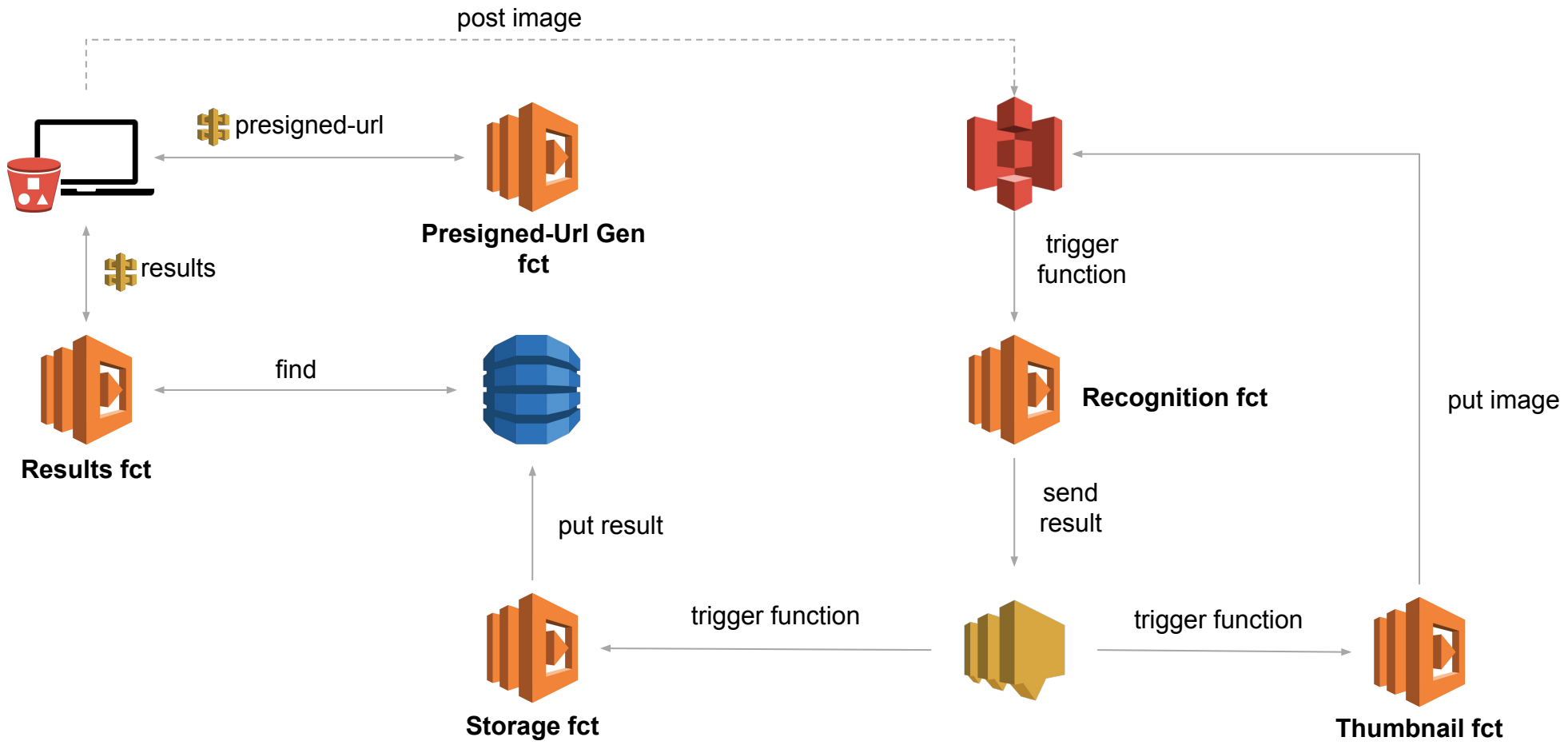
Code Deploy

MONITORING

X-Ray

Cloud Watch

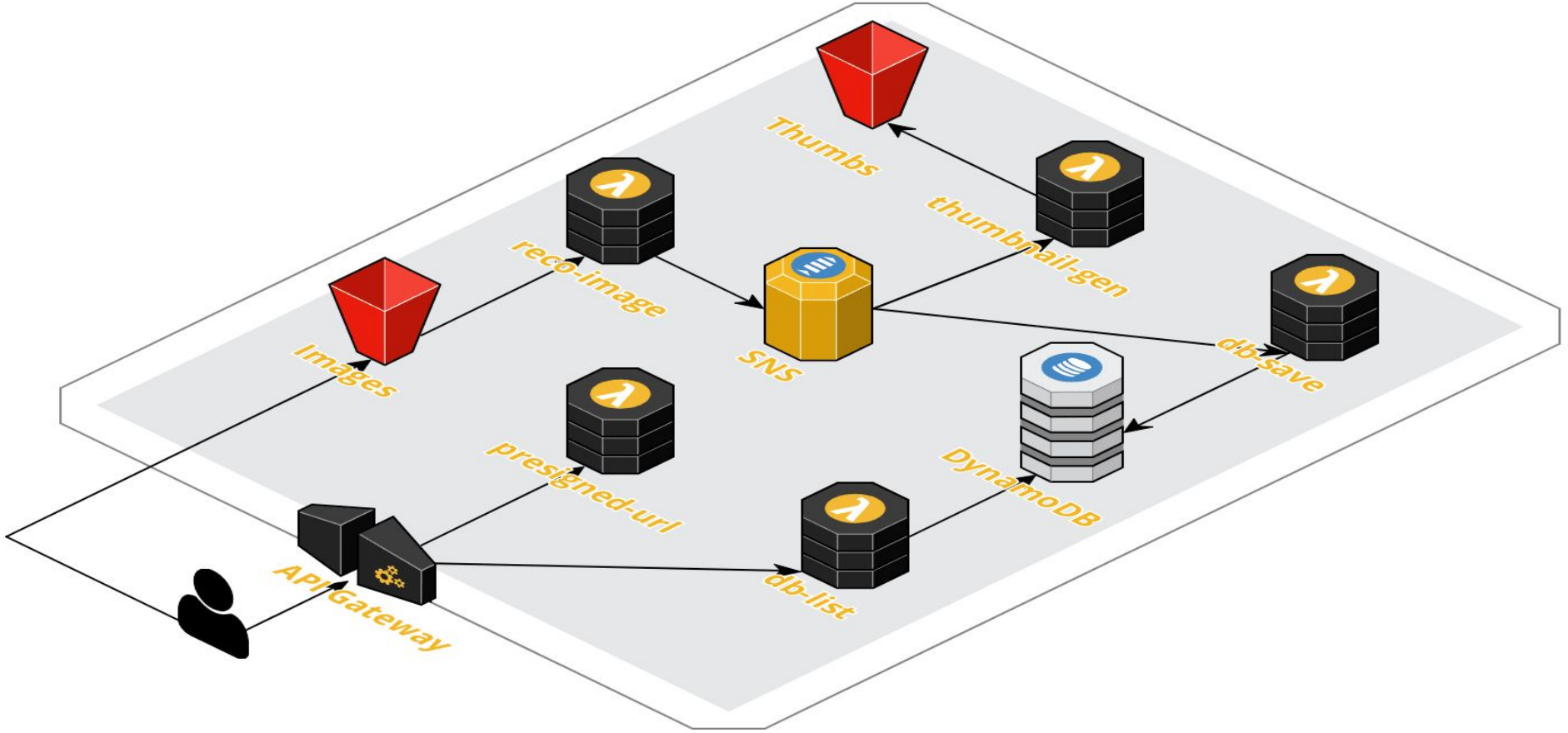
Hands-on Overview



Hands-on Overview

Compute	 Lambda
APIs	 API Gateway
Storage	 S3
Databases	 DynamoDB
Messaging	 SNS  SQS
Orchestration	 Step Functions  CloudWatch Events
Authentication	 Cognito

Hands-on - Logical Architecture



AWS Free Tier

COMPUTE

AWS Lambda

1 Million

free requests per month

Compute service that runs your code in response to events and automatically manages the compute resources

Learn more about AWS Lambda »

EXPAND DETAILS

APPLICATION SERVICES

Amazon API Gateway

1 Million

API Calls Received per month

Publish, maintain, monitor, and secure APIs at any scale

Learn more about Amazon API Gateway »

EXPAND DETAILS

STORAGE & CONTENT DELIVERY

Amazon S3

5 GB

of standard storage

Secure, durable, and scalable object storage infrastructure

Learn more about Amazon S3 »

EXPAND DETAILS

DATABASE

Amazon DynamoDB

25 GB

of storage

Fast and flexible NoSQL database with seamless scalability

Learn more about DynamoDB »

EXPAND DETAILS

MOBILE SERVICES

Amazon SNS

1 Million

publishes

Fast, flexible, fully managed push messaging service

Learn more about Amazon SNS »

EXPAND DETAILS

Planning

S3

Hands On #1 - S3 | IAM - Image bucket

API Gateway, Lambda

Hands On #2 - Lambda | API Gateway - Upload Image

Hands On #3 - Lambda - Image recognition

SNS : overview

Hands On #4 - SNS | Lambda - Publish Results

DynamoDB

Hands On #5 - DynamoDB - Save results

Extra Hands-On

Hands On #6 - API Gateway | Lambda - List items

Hands On #7 - S3 | Lambda - Generate thumbnail

Hands On #8 - S3 - Static website

S3

Simple Storage Service

S3 - Simple Storage Service

storage for the Internet

object storage service

bucket

unlimited storage

S3 - Benefits

scalability

data availability

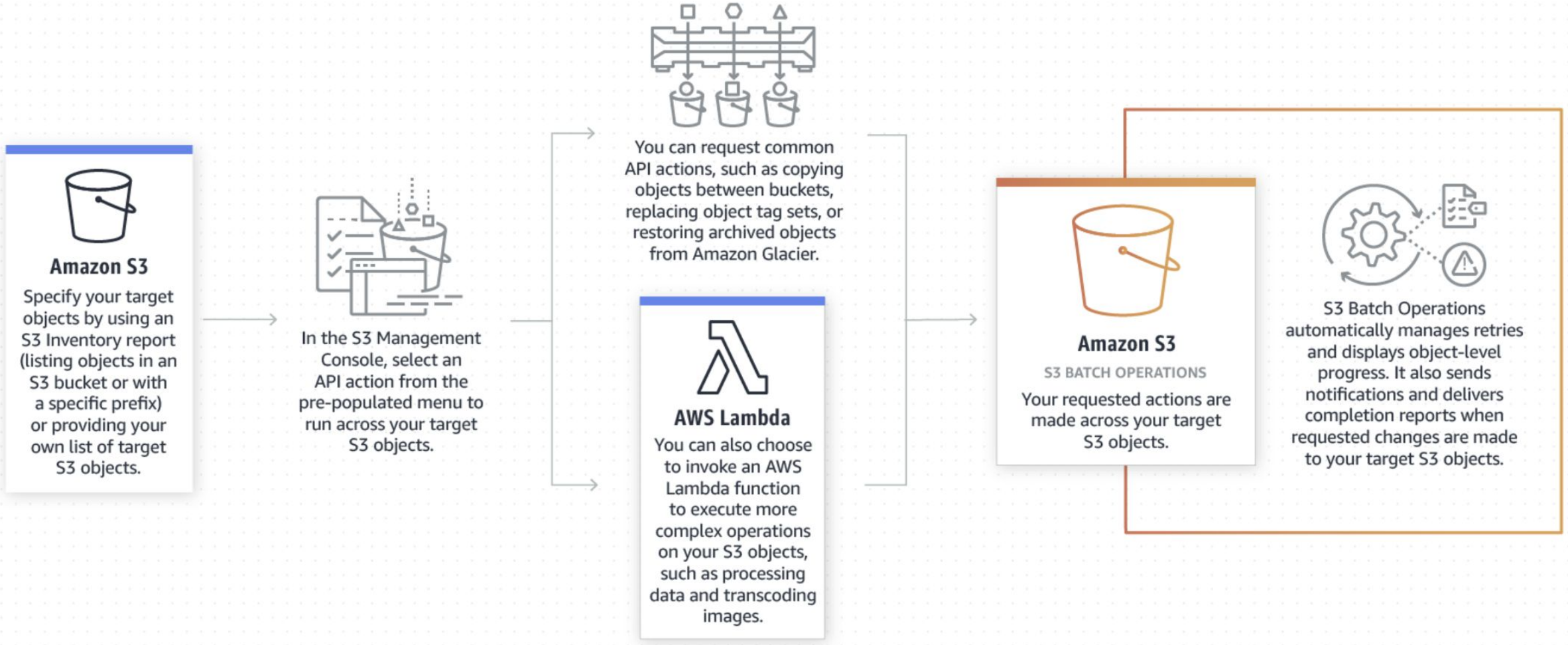
security

performance

S3 - Use cases

host static website
backup and restore
infrequent access
data archiving
online storage
file sharing
logs
datalake

S3 - How it works ?



S3 - Types

Standard
Intelligent Tiering
Standard Infrequent Access
IA One Zone

Glacier
Glacier Deep Archiving

S3 - Access

Console
SDK
CLI
API

S3 - Object

key

version id

value

metadata

access control information

S3 - Encryption

Server-Side Encryption

Client-Side Encryption

S3 - Pricing Model

per GB

per 1000 requests (GET, SELECT, PUT, COPY, POST, LIST)

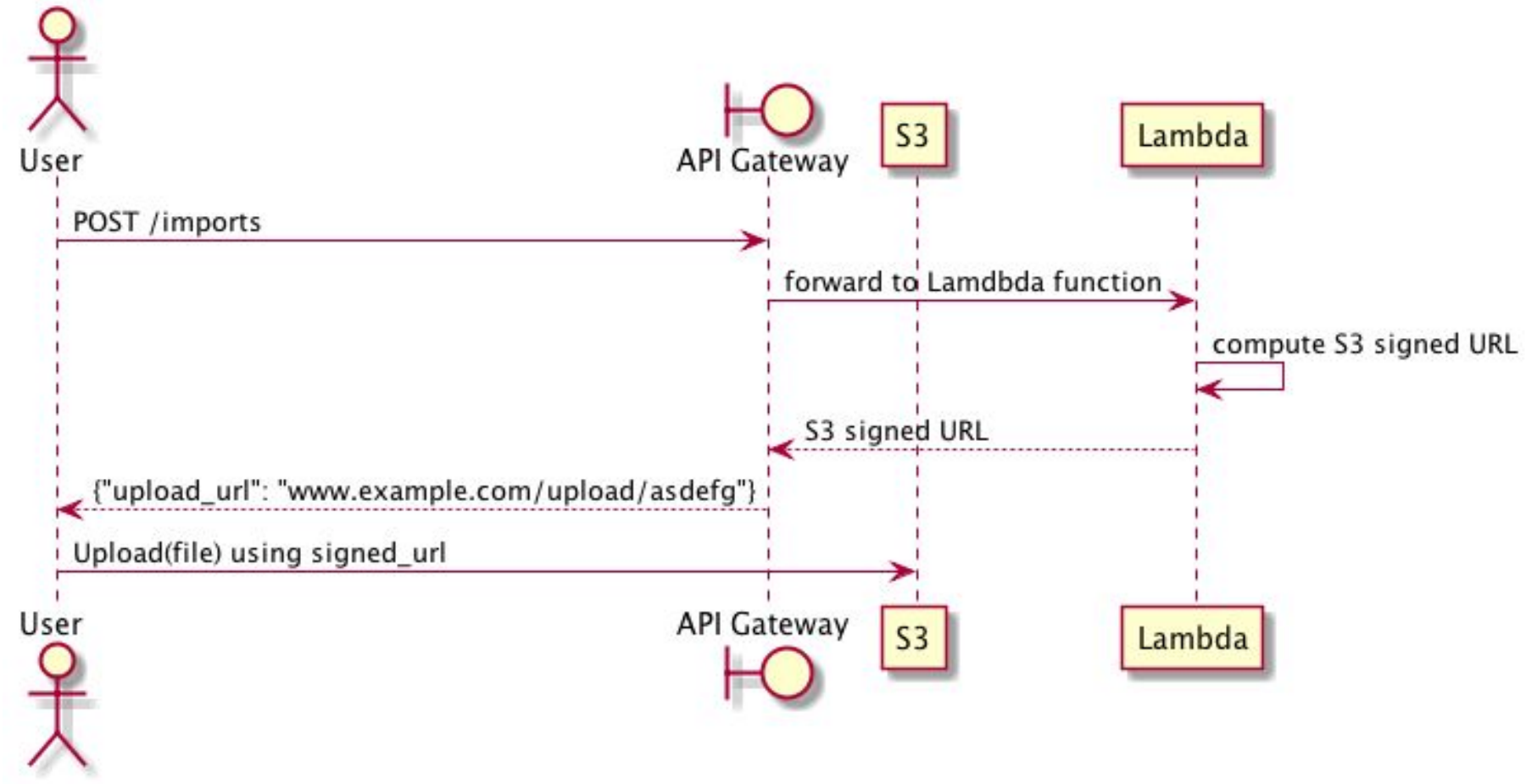
S3 - Presigned URLs

upload an object to a private bucket without having a AWS security credentials or permissions

generated programmatically using the AWS SDK:

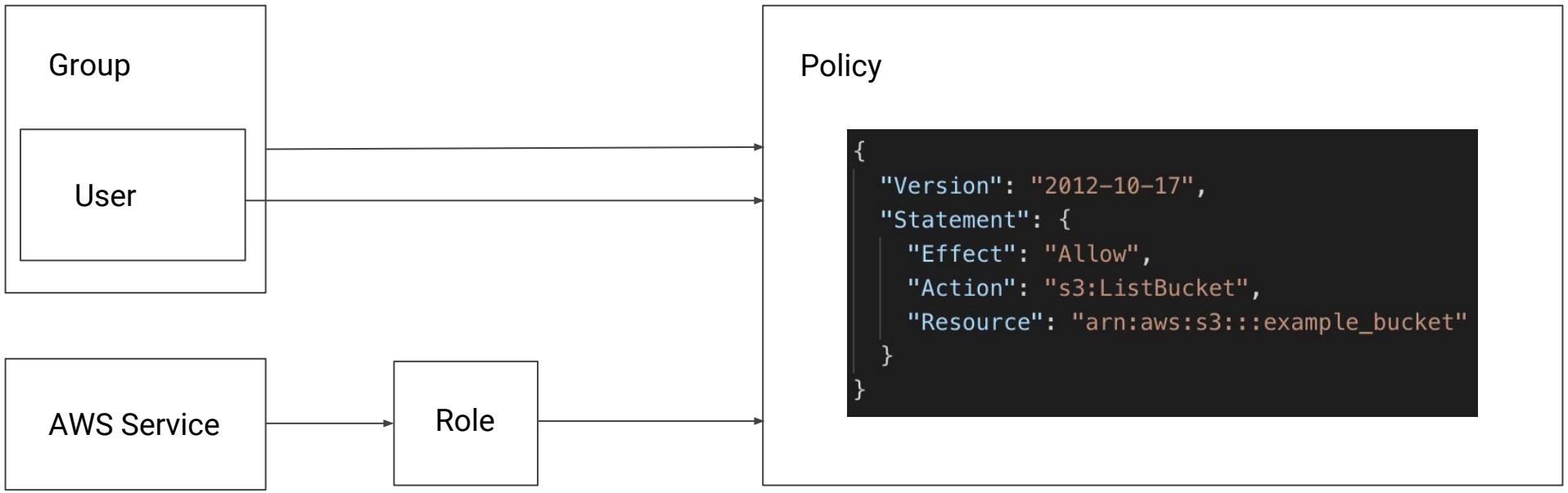
- security credentials
- bucket name
- object key
- HTTP method (PUT for uploading objects)
- expiration date and time

S3 - Presigned URLs

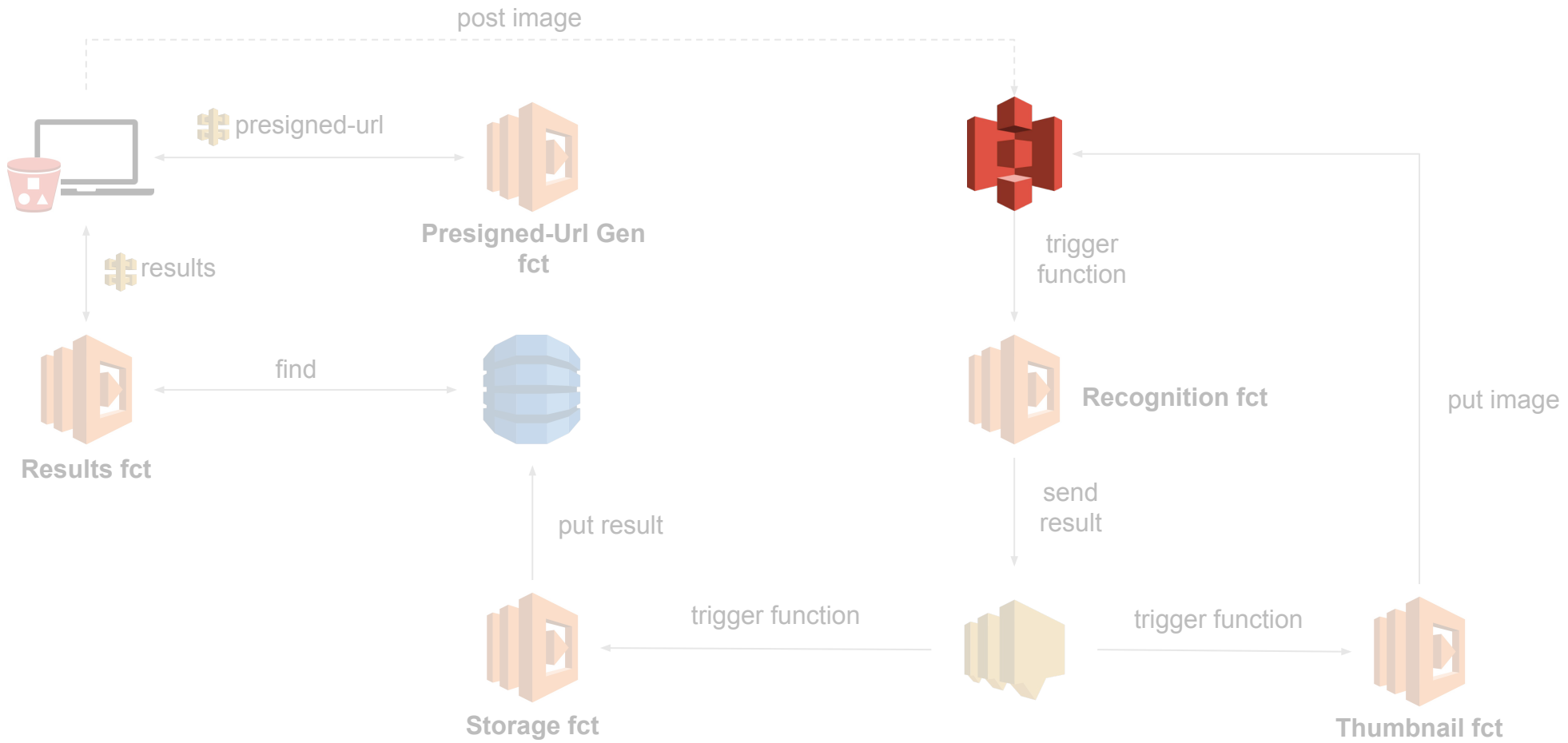


IAM

IAM - Policies Management



Hands-on Overview



S3 - Hands-on

https://bit.ly/2GypvoT

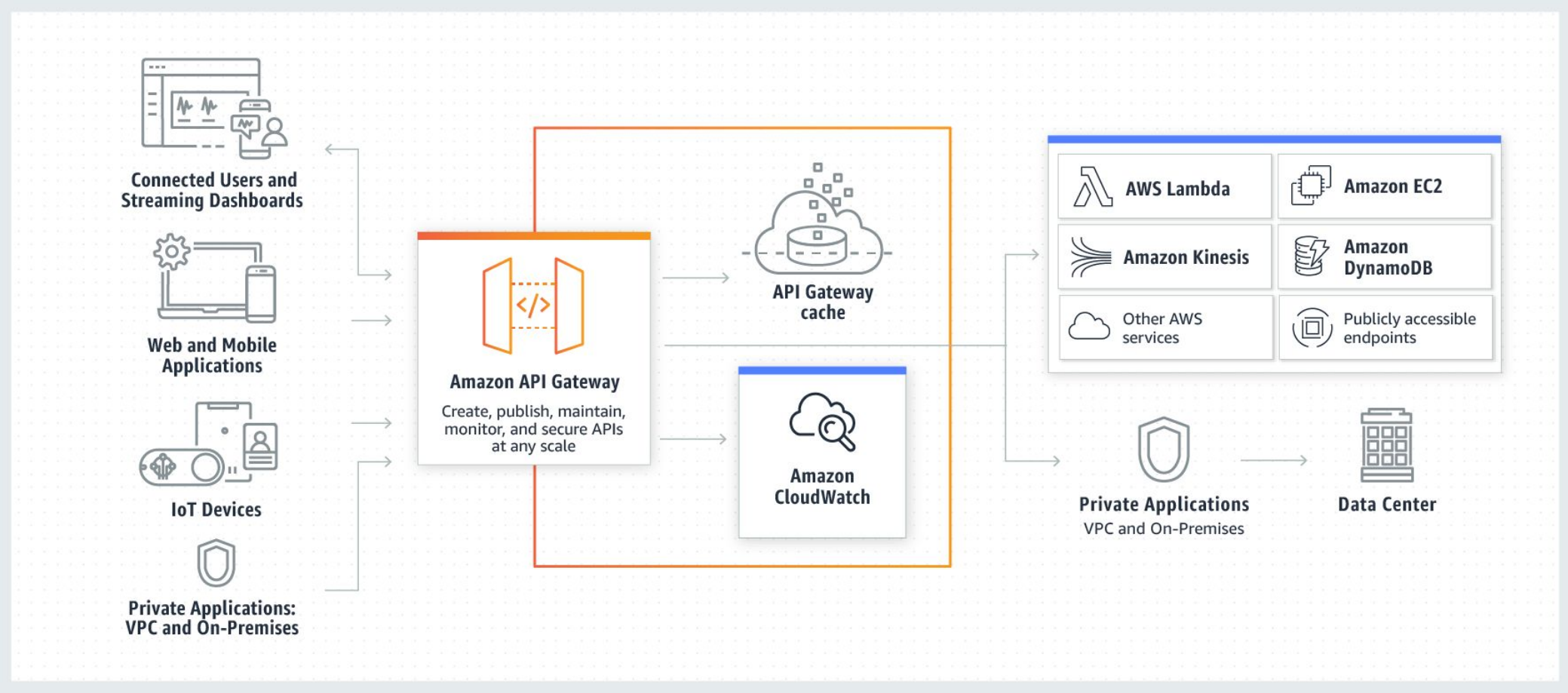
API Gateway

API Gateway

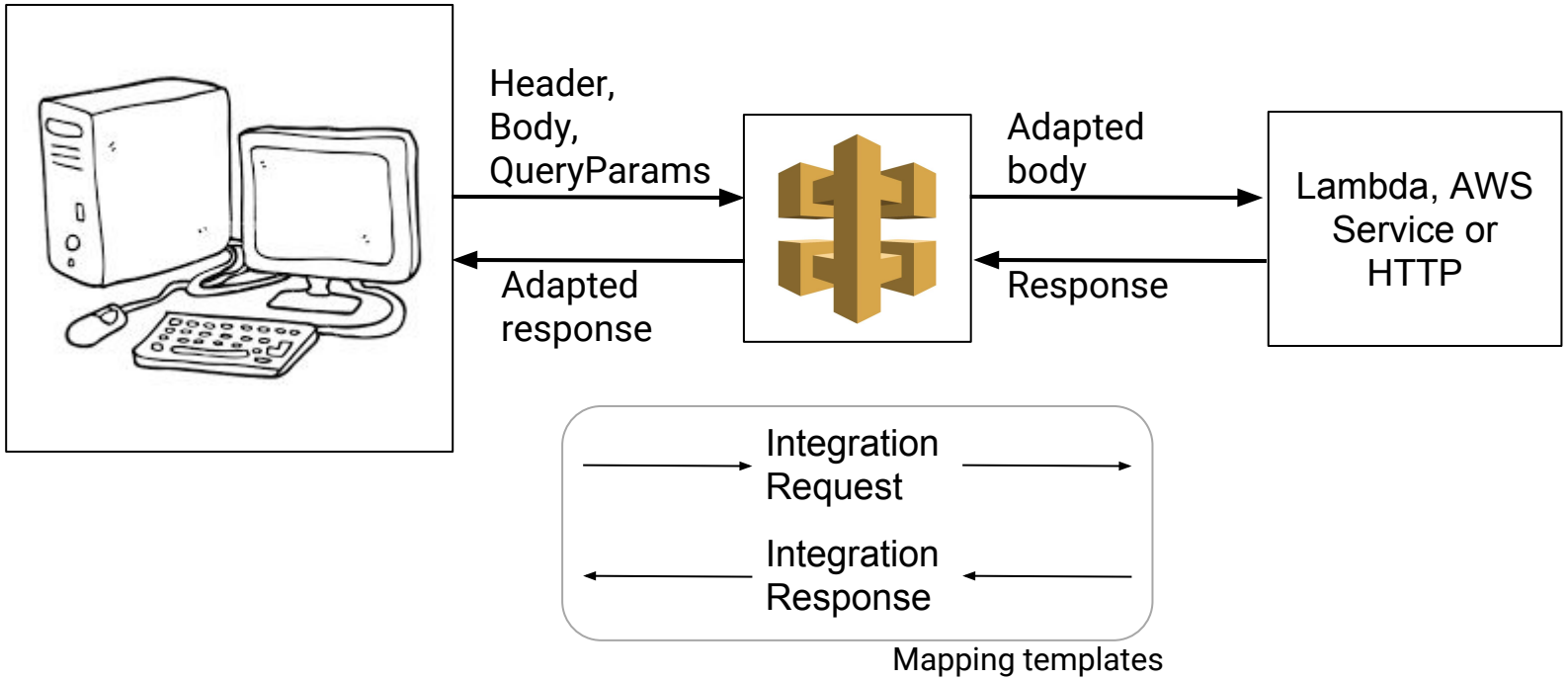
- create, publish, maintain, monitor, and secure APIs at any scale

- create REST and WebSocket APIs

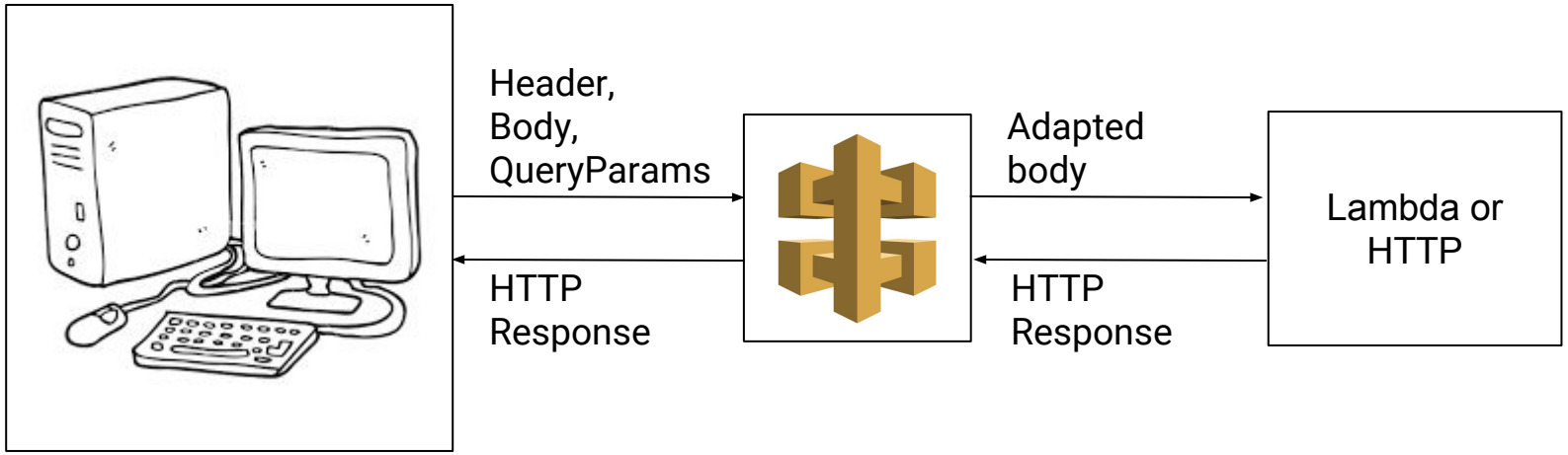
API Gateway - How it works ?



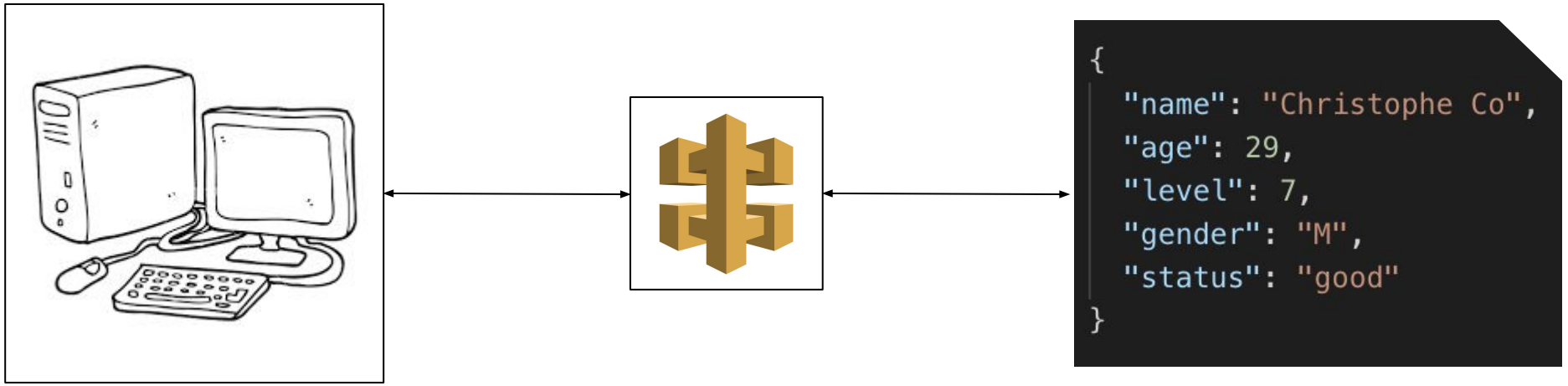
API Gateway - Mapping templates



API Gateway - Lambda or HTTP Proxy



API Gateway - Mock



API Gateway - Stages

- a logical reference to a lifecycle state
  for example : 'dev', 'prod', 'beta', 'v2'

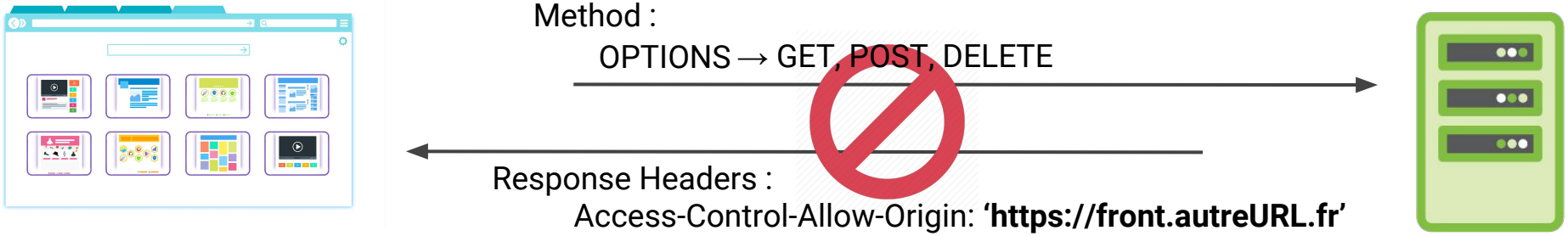
- API stages are identified by API ID and stage name.

API Gateway - Security

- resource policies
- lambda authorizers
- amazon Cognito
- client-side SSL certificates
- cross-origin resource sharing (CORS)

What are the CORS?

CORS : Cross Origin Resource Origin



Response headers

- Access-Control-Allow-Origin
- Access-Control-Allow-Credentials
- Access-Control-Expose-Headers
- Access-Control-Max-Age
- Access-Control-Allow-Methods
- Access-Control-Allow-Headers

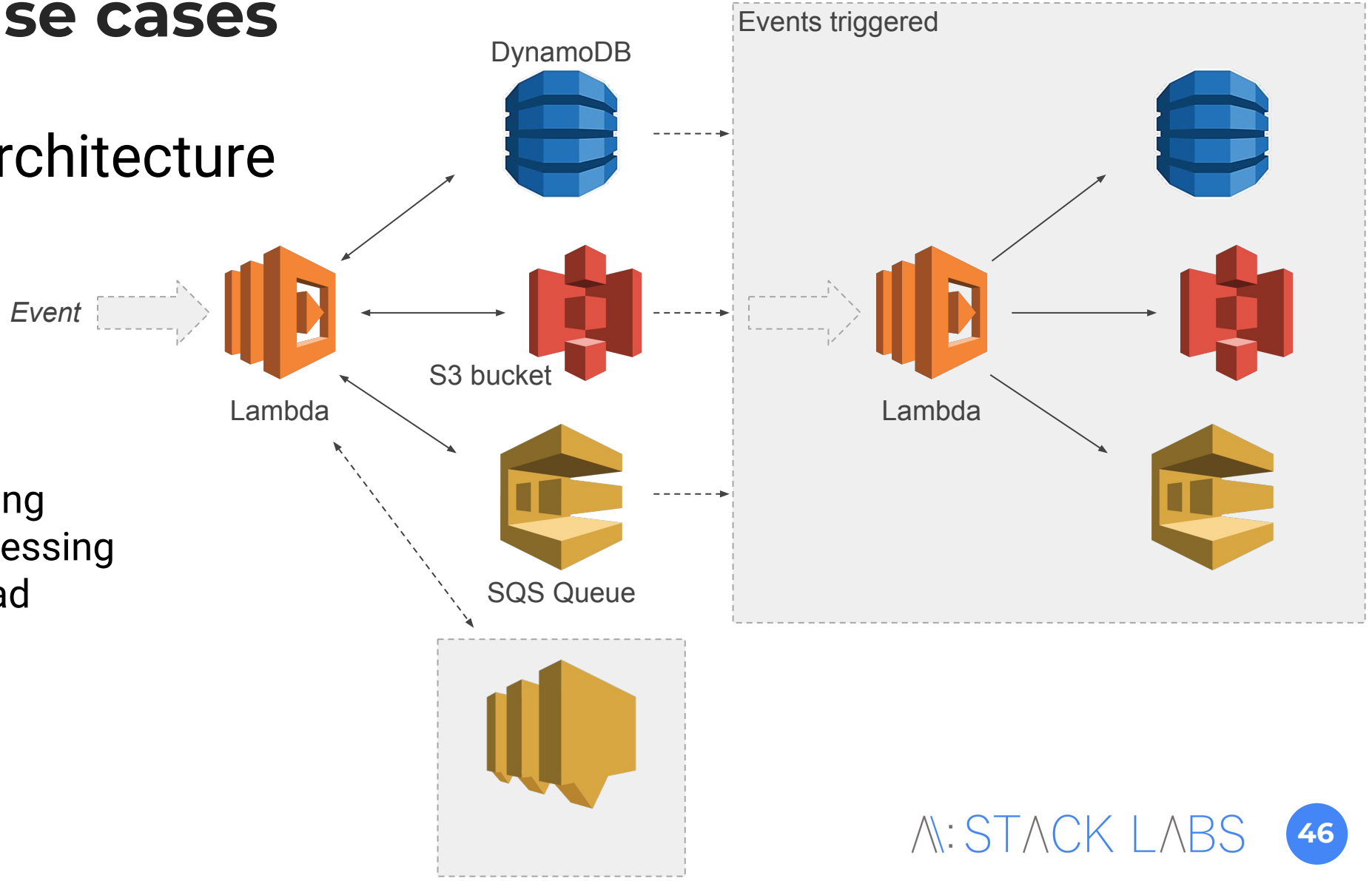
API Gateway - Pricing Model

- no minimum fees or startup costs.
- you pay only for the API calls you receive and the amount of data transferred out
- reduce your cost as your API usage scales

Lambda

Lambda - Use cases

Event-driven architecture



data processing
real-time file processing
real-time stream processing
extract, transform, load
backends
iot backends
mobile backends
web applications

Lambda - Event sources

Amazon S3
Amazon DynamoDB
Amazon Kinesis Data Streams
Amazon Simple Notification Service
Amazon Simple Email Service
Amazon Simple Queue Service
Amazon Cognito
AWS CloudFormation
Amazon CloudWatch Logs
Amazon CloudWatch Events
AWS CodeCommit
AWS Config
Amazon Alexa
Amazon Lex
Amazon API Gateway
AWS IoT Button
Amazon CloudFront
Amazon Kinesis Data Firehose

Lambda - Benefits

- no servers to manage
- continuous scaling
- subsecond metering (you are charged for every 100ms your code executes and the number of times your code is triggered)

Lambda - Languages

- Java
- Go
- PowerShell
- Node.js
- C#
- Python
- Ruby

Lambda - Pricing Model

- pay only for the compute time you consume
- there is no charge when your code is not running.

Lambda - Limits

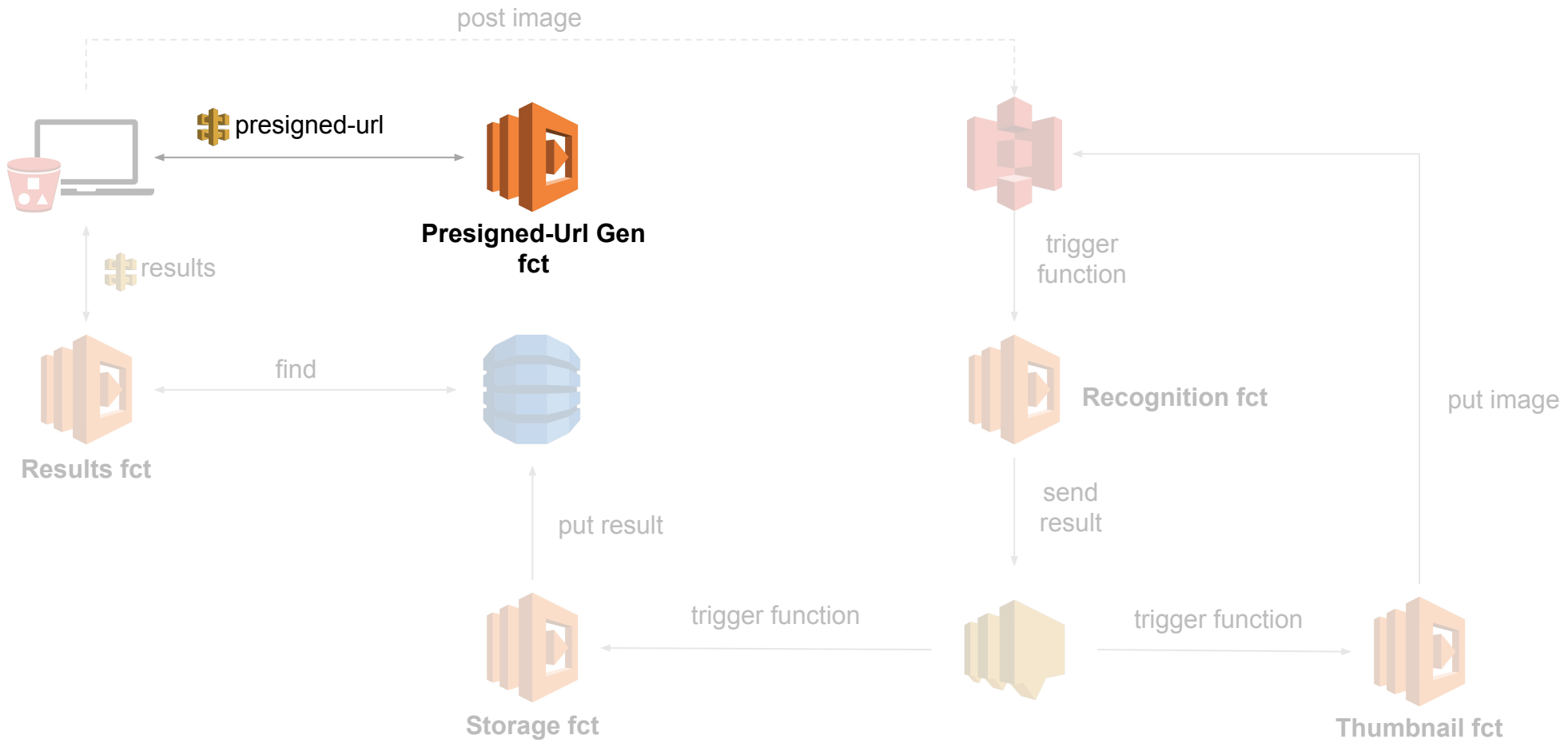
Resource	Limit
Function memory allocation	128 MB to 3008 MB, in 64 MB increments.
Function timeout	900 seconds (15 minutes)
Function environment variables	4 KB
Function resource-based policy	20 KB
Function layers	5 layers
Invocation payload (request and response)	6 MB (synchronous) 256 KB (asynchronous)
Deployment package size	50 MB (zipped, for direct upload) 250 MB (unzipped, including layers) 3 MB (console editor)
Test events (console editor)	10
`/tmp` directory storage	512 MB
File descriptors	1024
Execution processes/threads	1024

Lambda in console

Let’s Go To The Console

https://aws.amazon.com/console

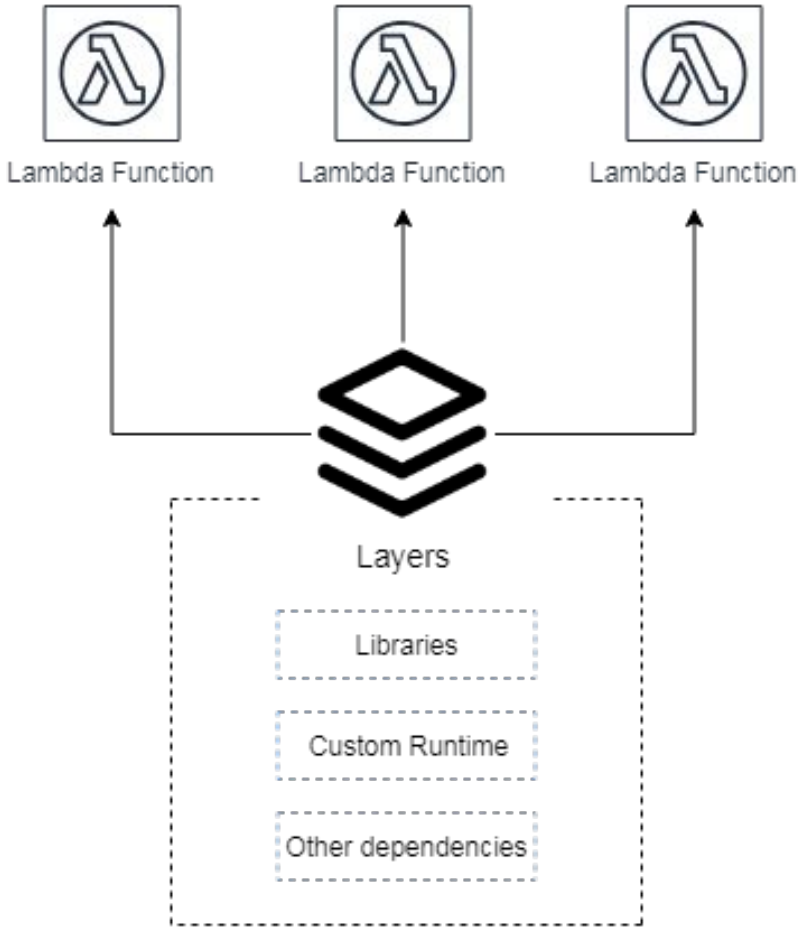
Hands-on Overview



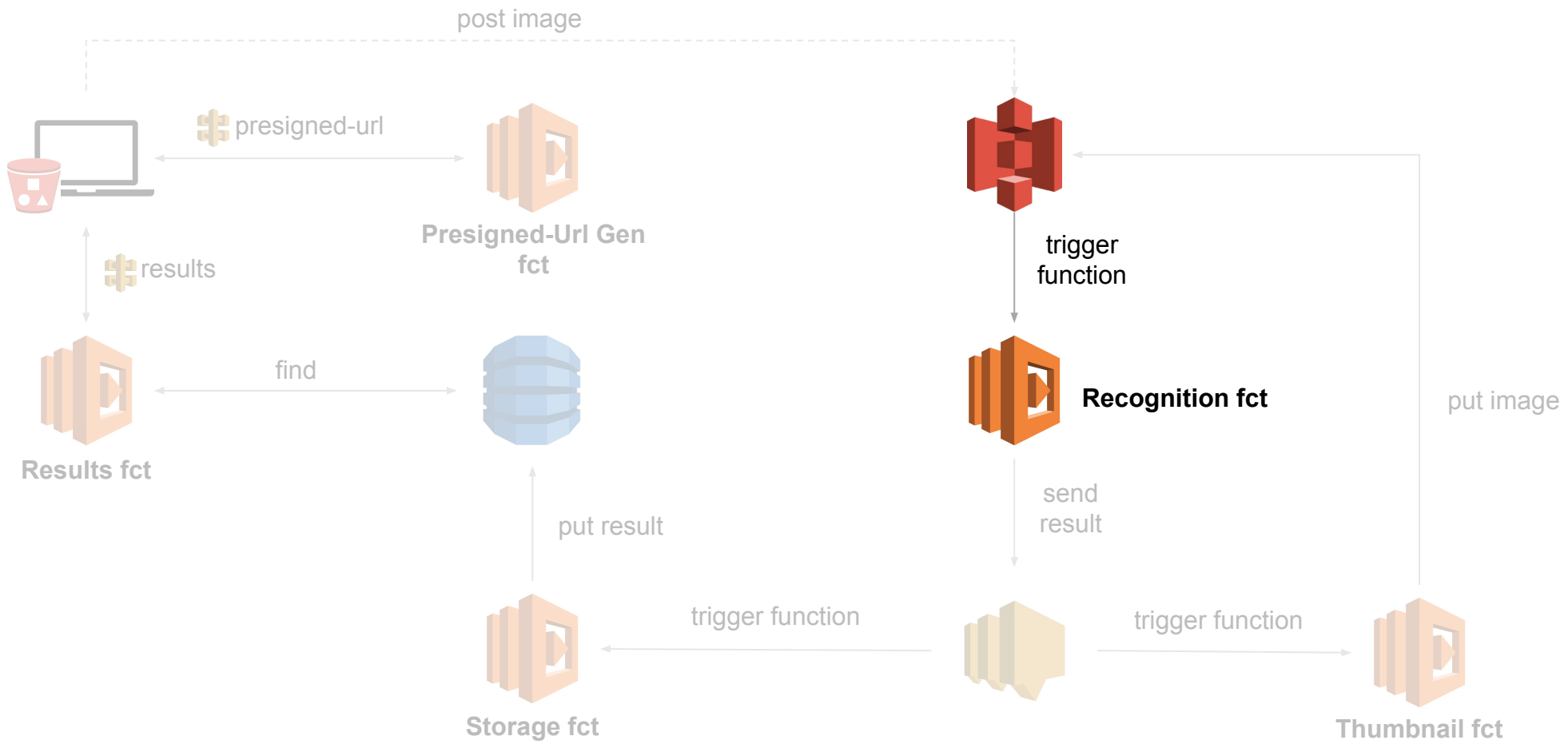
Lambda - Presigned Url - Hands-on

https://bit.ly/2I1Es5g

Lambda - Layers



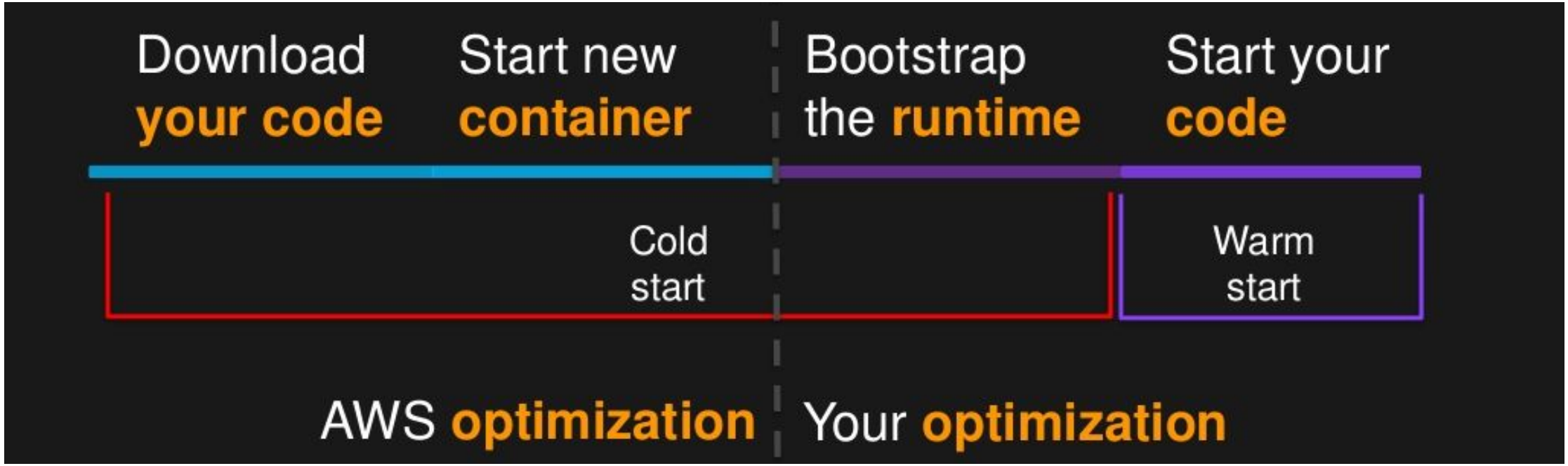
Hands-on Overview



Lambda - Image Reco - Hands-on

https://bit.ly/2DklqSb

Lambda - Cold Start & Warm Start



Source: AWS re:Invent 2017 - Become a Serverless Black Belt

Lambda - Cold Start & Warm Start

```python
import boto3
from io import BytesIO
import json
import os

s3 = boto3.client('s3')

def lambda_handler(event, context):

    # some code

    return "Done"
```

Cold Start: code run at initialisation only

Warm Start: code run every time

Lambda in console

Let’s Go To The Console

https://aws.amazon.com/console

SNS

Simple Notification Service

SNS - Simple Notification Service

topics

push-based

decouple microservices, distributed systems, serverless applications

SNS - Benefits

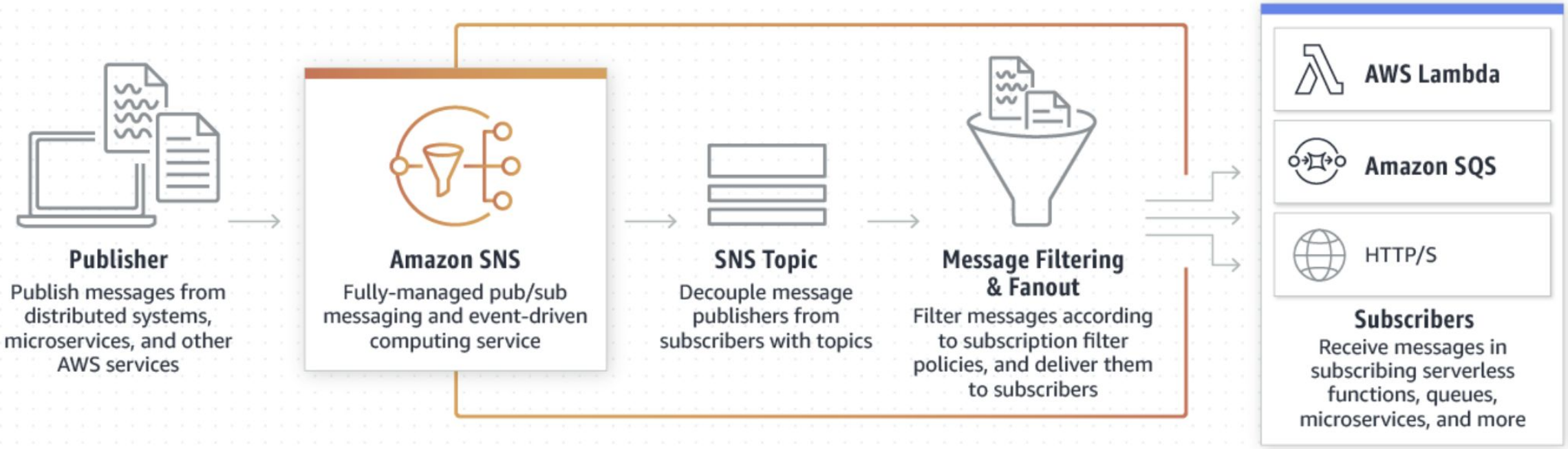
highly available

durable

secure

pub/sub messaging service

SNS - How it works ?



SNS - JSON Format

Message
TopicArn
Type
...

SNS - Notification deliveries

Mobile Push Notifications
SMS
Email
HTTP/s
Simple Queue Service (SQS)
Lambda functions

SNS - Event Source and Destinations

event sources

- DynamoDB
- S3
- Lambda
- +28

event destinations

- Lambda
- SQS

SNS - Encryption

AWS Key Management Service

SNS - Access

Console
SDK
CLI
API

SNS - Pricing Model

per 1 million mobile push notifications

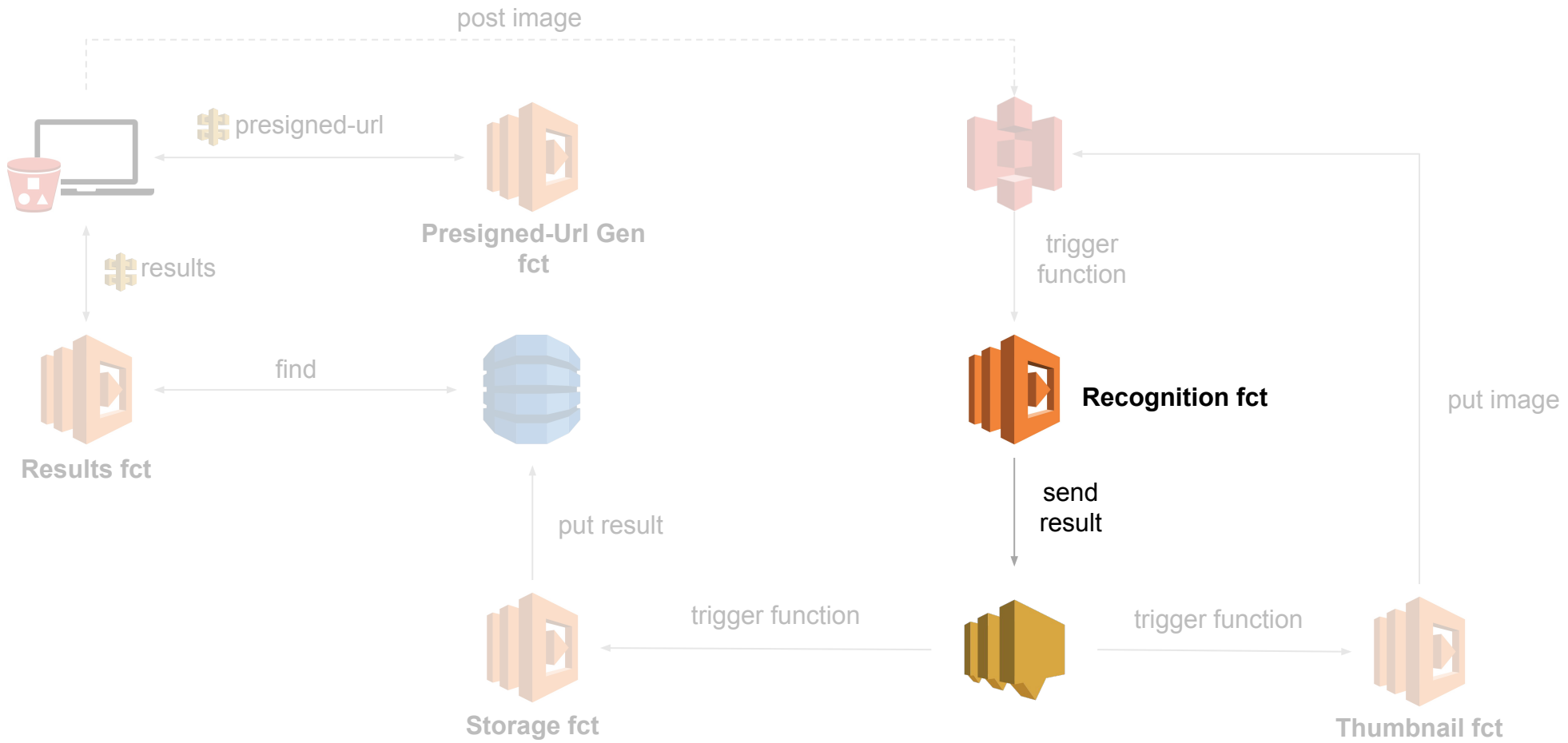
per 100,000 email

per million HTTP/s

no charge for deliveries to Lambda & SQS Queues

data transfer IN (free)/OUT per GB

Hands-on Overview



SNS - Create Topic - Hands-on

https://bit.ly/2WLYx37

DynamoDB

What is DynamoDB ?

NoSQL Database

Key-value and document oriented

Provisioned throughput model

Accessible via simple webservice APIs

DynamoDB Benefits


Fully managed


Fast, consistent performance


Highly scalable


Flexible


Event-driven programming


Fine-grained access control

Consistency models

Eventually consistent

Strongly consistent

Provisioned throughput model

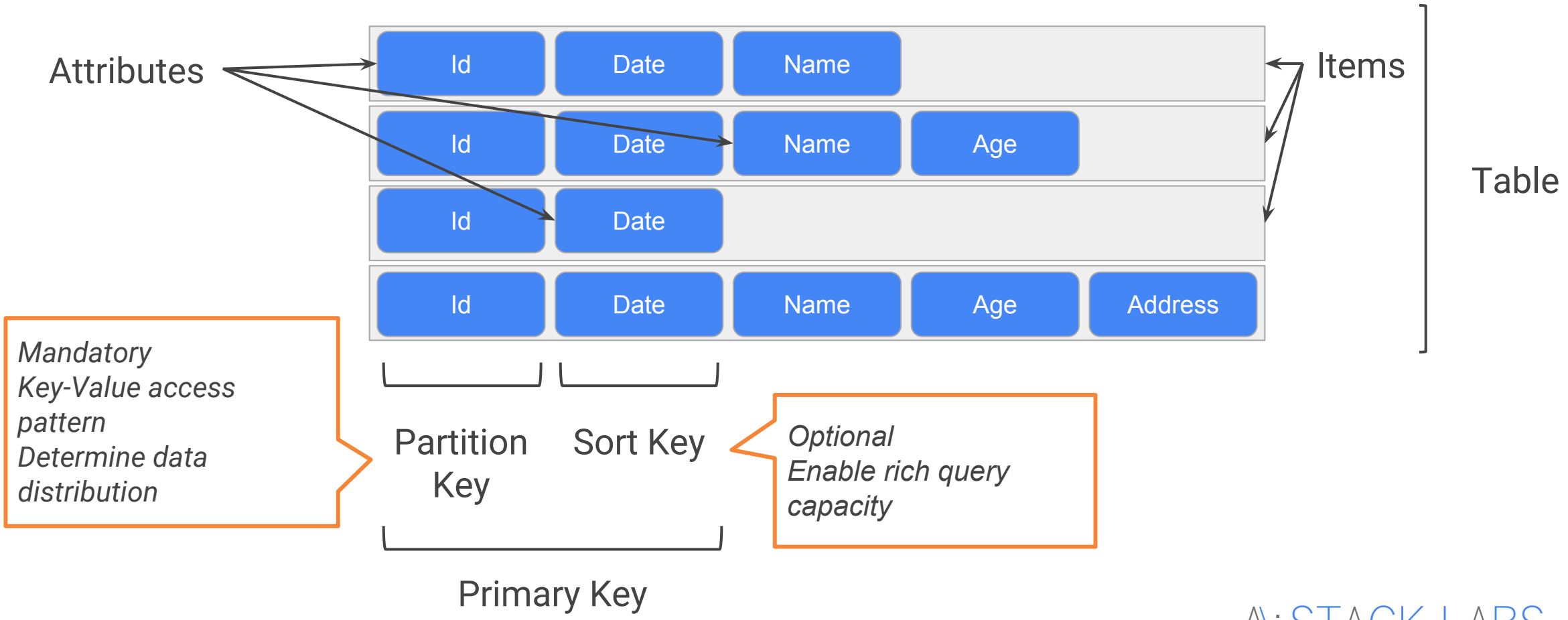
Read capacity unit:

1*Read capacity = 1*4KB Read / sec. (eventually consistent)
1*Read capacity = 2*4KB Read / sec. (strongly consistent)

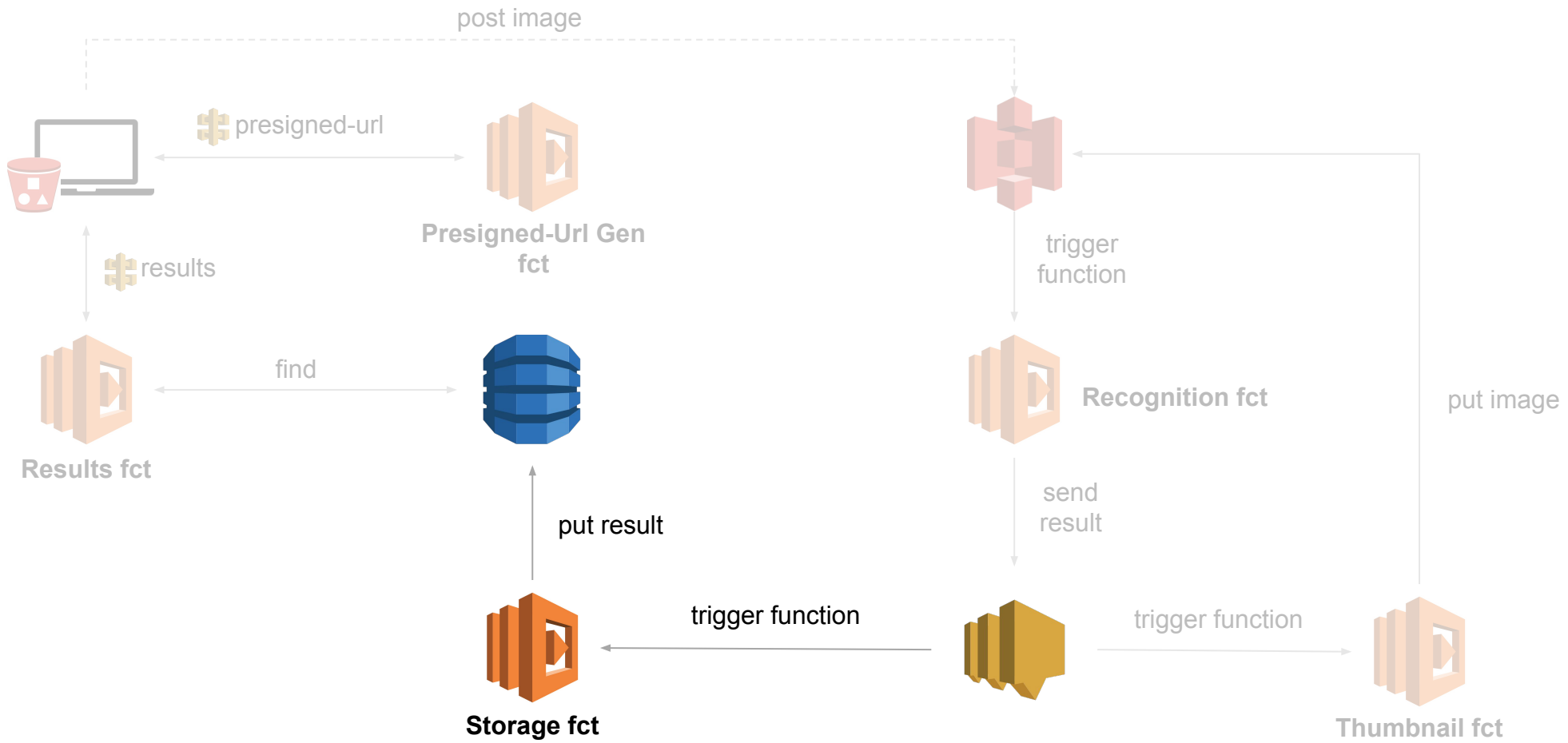
Write capacity unit:

1*Write capacity = 1*1KB Write / sec.

DynamoDB Table Structure



Hands-on Overview

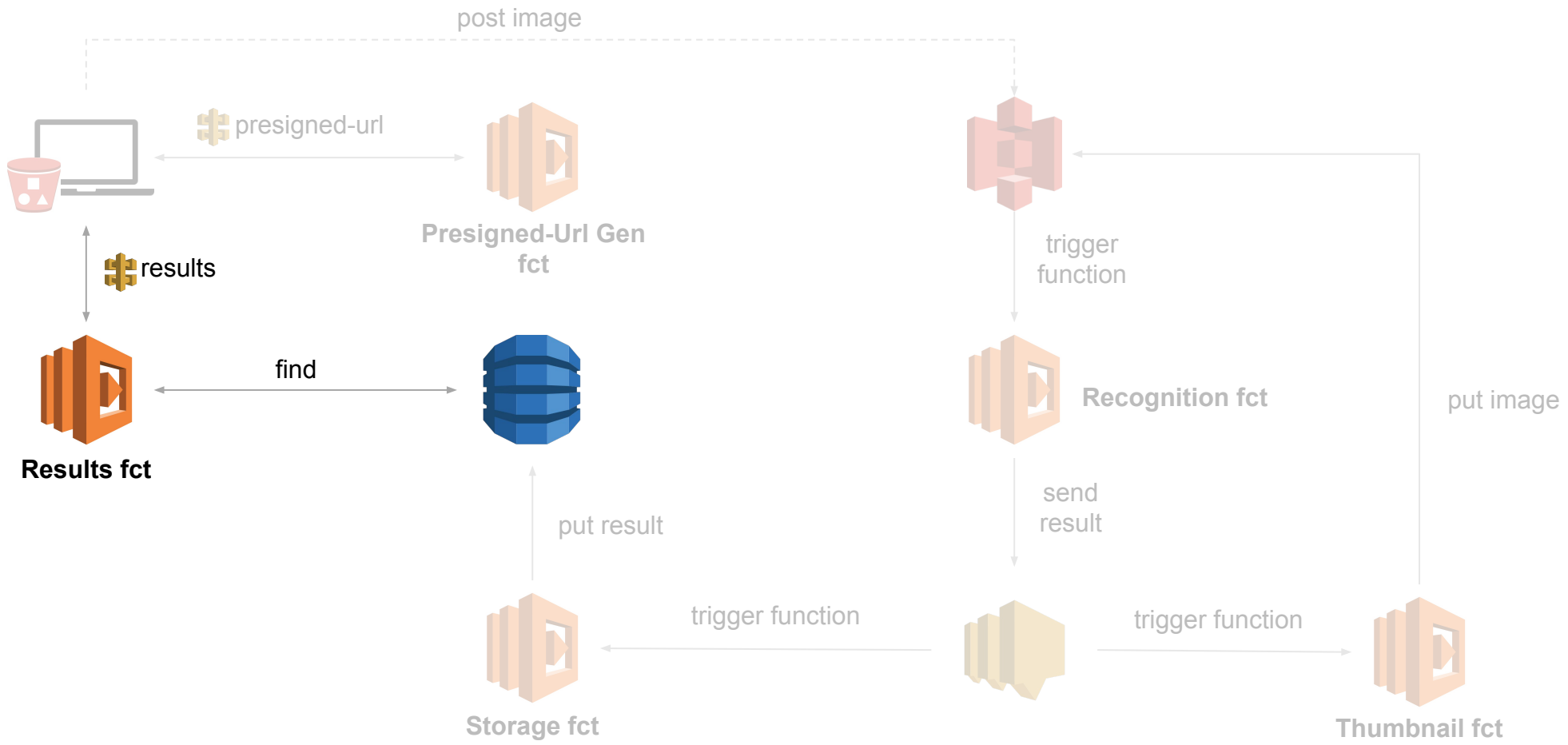


DynamoDB - Save Result - Hands-on

https://bit.ly/2MUgZC0

Extra Hands-on

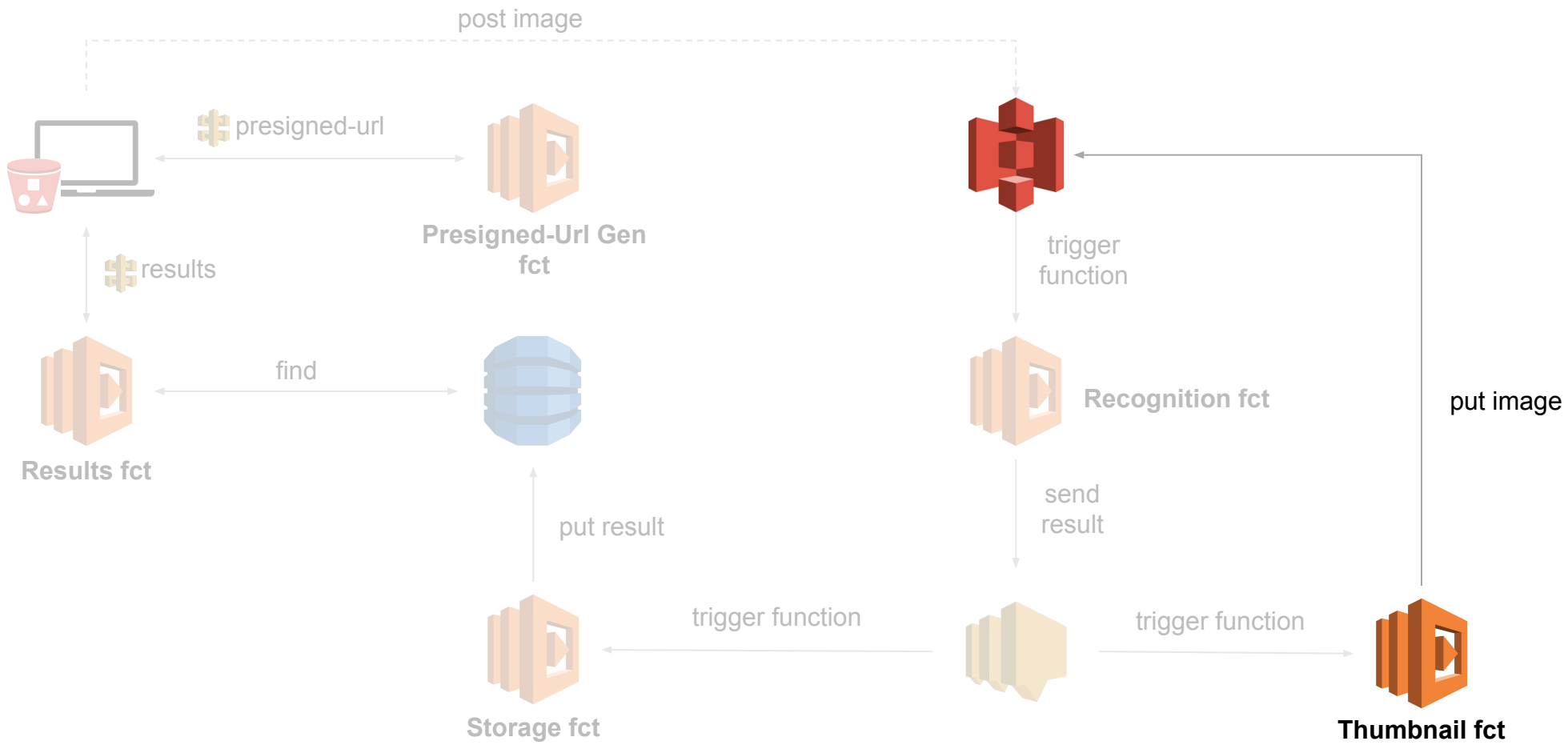
Hands-on Overview



DynamoDB - List Results - Hands-on

https://bit.ly/2SztRTn

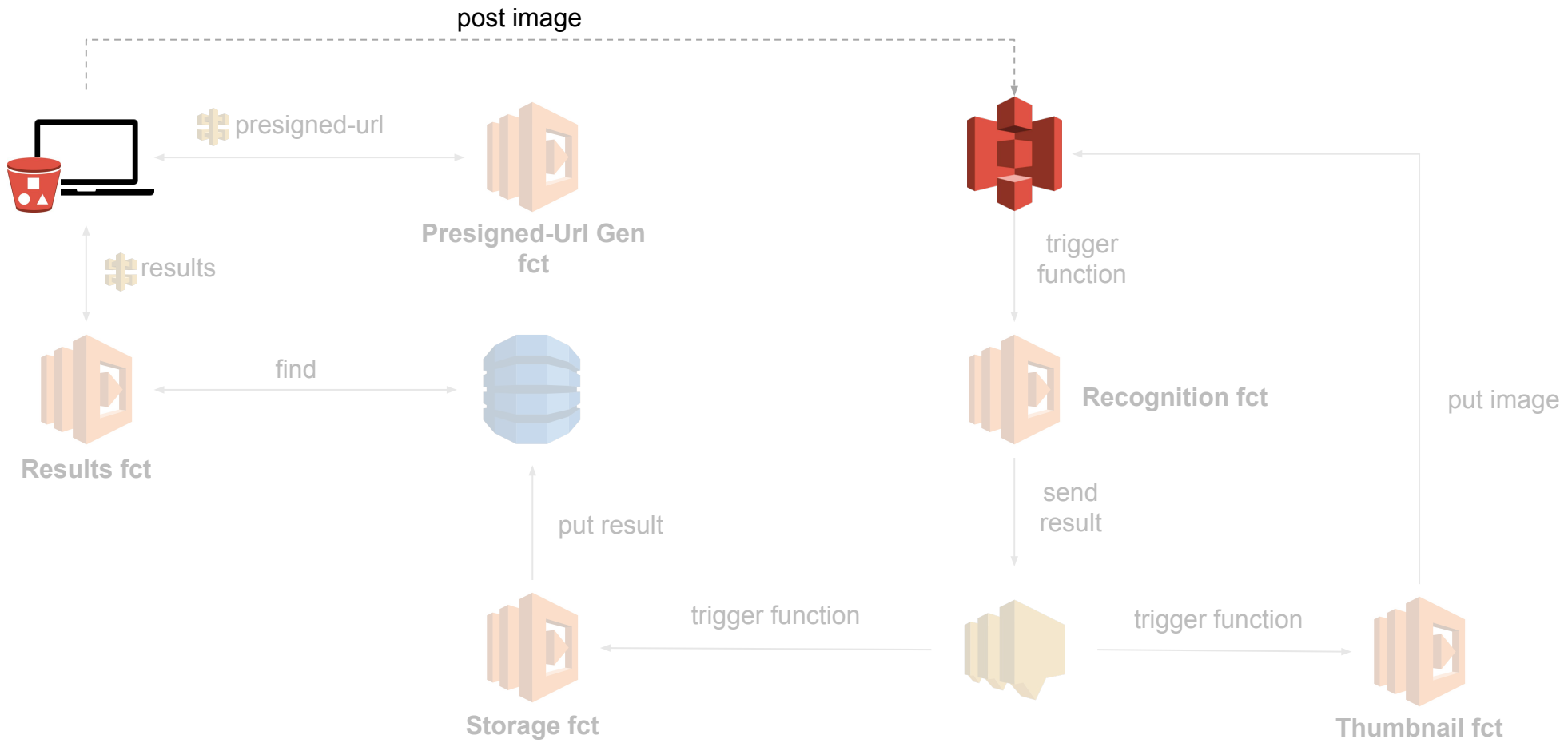
Hands-on Overview



Lambda - Thumbnail creation - Hands-on

https://bit.ly/2DkSbPf

Hands-on Overview



S3 - Static Webhosting - Hands-on

https://bit.ly/2S4Vorw

Clean-Up

Minimum Clean-Up

API Gateway:

1. Open the API Gateway created during Hands-On
2. Delete the stage “dev”

S3:

1. Open the thumbnail S3 bucket
2. Remove all public access and policy

Full Clean-Up

Delete the following components:

1. API Gateway created during hands-on
2. Following Lambda functions:
    - presigned-url
    - reco-image
    - db-save
    - db-list
    - thumbnail-gen
3. DynamoDB table
4. SNS Topic

THANKS!

Blog https://blog.stack-labs.com/

https://stack-labs.com/

References

- https://d1.awsstatic.com/whitepapers/serverless-architectures-with-aws-lambda.pdf
- https://medium.com/devopslinks/dry-in-serverless-with-aws-lambda-layers-1d8cd847caa4
- https://aws.amazon.com/fr/free/?awsf.Free%20Tier%20Types=categories%2312monthsfree