

# Hands-on Cloud Computing Services

## Lezione 5

Gabriele Russo Russo  
*University of Rome Tor Vergata, Italy*

A.A. 2023/24



**TOR VERGATA**  
UNIVERSITÀ DEGLI STUDI DI ROMA

# Simple Queue Service (SQS)

- ▶ Fully managed queueing service
- ▶ Enables decoupled communication among microservices/components
- ▶ Developers can avoid spending effort on a communication middleware
- ▶ Standard queues (at-least-once)
- ▶ FIFO queues (exactly-once, FIFO order)
  
- ▶ `producer.py` e `consumer.py`

## Idea: Photogallery + SQS

- ▶ New images uploaded to S3 in the `pending/` directory
- ▶ Image processing (resizing, filters,...) delegated to *workers*
- ▶ Web server and workers communicate through SQS (decoupled)

# AWS Lambda

- ▶ Function-as-a-Service offering by AWS
- ▶ Enables the execution of serverless functions
- ▶ Functions can be written using many different languages
- ▶ Fast scaling from zero to “infinity”
- ▶ Fine-grained pay-per-use pricing



## Lambda Invocation (2)

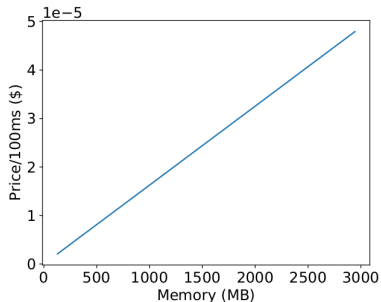
Lambda functions can be invoked in several ways, including:

- ▶ AWS CLI
- ▶ AWS SDK (boto)
- ▶ HTTP(S) endpoints ([discussed in the next lecture](#))
- ▶ automatically in response to events (e.g., new upload to S3)

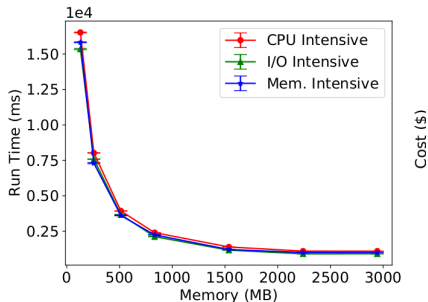
# AWS Lambda: Hello World

- ▶ Let's create our first Lambda instance
- ▶ We can start from a *blueprint*: "Hello, world!"
- ▶ We can create **Test** events for our function
- ▶ Test the function: observe duration, billed duration, and init duration
- ▶ **Cold start**
- ▶ We can invoke the function using the SDK and the CLI
  - ▶ e.g., `invoke_hello_world.py`

# AWS Lambda: Sizing



(a) AWS price per 100ms

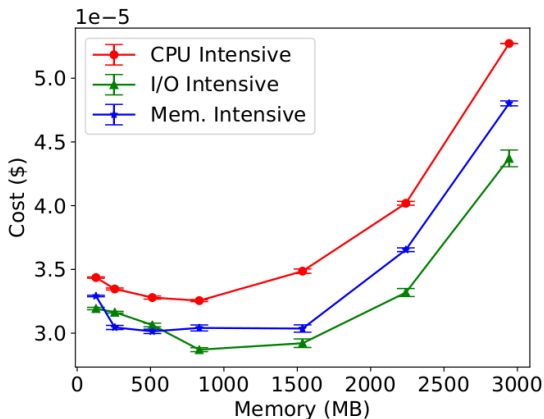


(b) Run-time of serverless functions on Amazon Lambda

Source: Nabeel Akhtar, Ali Raza, Vatche Ishakian, Ibrahim Matta: **COSE: Configuring Serverless Functions using Statistical Learning**. INFOCOM 2020: 129-138



## AWS Lambda: Sizing (2)



(c) Cost for running functions on Amazon Lambda

Source: Nabeel Akhtar, Ali Raza, Vatche Ishakian, Ibrahim Matta: **COSE: Configuring Serverless Functions using Statistical Learning**. INFOCOM 2020: 129-138

# Invoking Lambda functions via HTTP

- ▶ We often want to expose serverless functions as HTTP services (e.g., as part of a larger REST/HTTP API)
- ▶ Two possible approaches to do so (each with pros and cons)
- ▶ **Solution 1:** Lambda function URLs
- ▶ **Solution 2:** AWS API Gateway

# Function URLs

- ▶ New feature launched in 2022
- ▶ You can associate an HTTP URL to a Lambda function
- ▶ Requests targeted to the URL served by the function
- ▶ You can enable it using the web UI or Terraform

## Example: Lambda + Function URL

- ▶ Toy example: a Pow function
  - ▶ Handles GET requests with parameters  $x$  and  $y$
  - ▶ Returns  $x^y$
- ▶ We want a Lambda function to handle HTTP requests
- ▶ Source code: `lambda_pow_url`

# AWS API Gateway

- ▶ AWS Service to create, publish and maintain REST, HTTP and WebSocket APIs
- ▶ Developers create APIs integrated with other web or AWS services (EC2, Lambda, ...).
- ▶ Supports authentication mechanisms and fine-grained API management.
- ▶ Pricing: ~ 1 \$ per 300M API calls

# API Gateway: Key Concepts

- ▶ API (e.g., `mydomain.com/app/`)
- ▶ Deployment (e.g., test, stage, production)
- ▶ Resource (e.g., `/app/posts/`)
- ▶ Methods (e.g., `/app/posts/` → GET)
- ▶ Integration (e.g., Lambda functions, custom HTTP backends)

## Example: Lambda + API Gateway

- ▶ Let's integrate Lambda and API Gateway
- ▶ We want Lambda functions to handle HTTP requests
- ▶ Same example: Pow function
- ▶ Source code: `lambda_pow`

## API Gateway + CORS

- ▶ When calling your APIs from a frontend you will likely encounter issues related to CORS (Cross-Origin Resource Sharing)
- ▶ <https://docs.aws.amazon.com/apigateway/latest/developerguide/how-to-cors.html>
- ▶ When using the Lambda Proxy integration, it suffices to return the correct headers in the Lambda response (as we did for the previous function)
- ▶ **Example:** web frontend for pow



## Function URLs vs API Gateway

	<b>Function URL</b>	<b>API Gateway</b>
API Type	HTTP	HTTP, WebSocket
Authentication	IAM	IAM, AWS Cognito, API Key
Custom Domain	(through CloudFront)	Yes
Caching	No	Yes
Resp. Timeout	15 min	29 sec
Integrations	Lambda	Lambda, EC2, ...
Extra Price	No	Yes

# Serverless Image Processing

We need to process user-provided images for our social network.

For each image, we receive a URL in input and we need to:

- ▶ detect objects in the image (e.g., for moderation)
- ▶ create a thumbnail for fast preview
- ▶ create a grayscale version of the image

# Serverless Image Processing: Solution 1

- ▶ We define a Lambda function to solve the whole problem
- ▶ Source code: `lambda_image/`

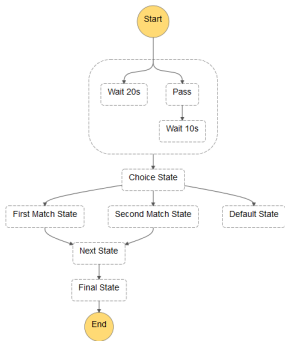
# Issues with Solution 1

- ▶ Cannot use different languages for different sub-tasks
- ▶ Cannot reuse sub-task functionalities (e.g., thumbnail creation)
- ▶ Sub-tasks not easily performed in parallel (e.g., grayscale conversion and object detection)

# Step Functions

A service for [serverless function orchestration](#).

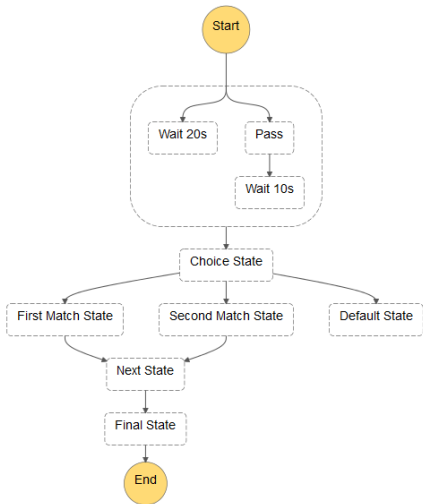
- ▶ Workflow definition by means of **state machines**
- ▶ Amazon States Language (JSON)
- ▶ Pricing: \$0.025 per 1,000 state transitions



## Step Functions (2)

Various types of state:

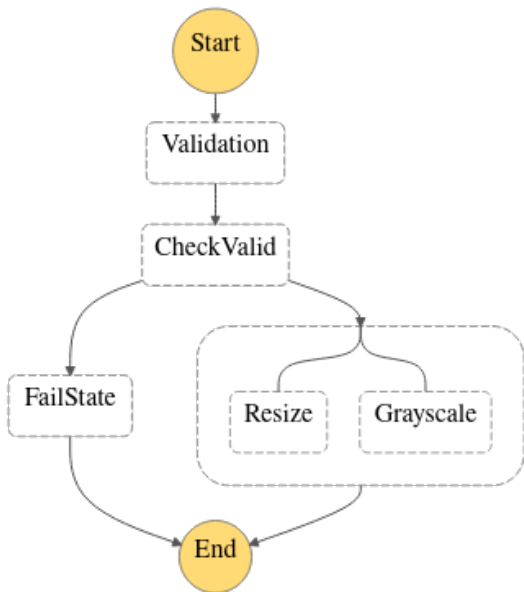
- ▶ Task (executes a Lambda function, or a different *activity*)
- ▶ Choice
- ▶ Fail
- ▶ Wait
- ▶ Parallel
- ▶ Map



## Serverless Image Processing: Solution 2

- ▶ We rely on 3 Lambda functions:
  - ▶ CheckImage()
  - ▶ Resize()
  - ▶ Grayscale()
- ▶ We exploit Step Functions for orchestration
- ▶ Source code: `lambda_image2`

## Serverless Image Processing: Solution 2





## Lambda + EFS

- ▶ Our functions relied on S3 to store the image throughout application execution
- ▶ S3 not well suited for low-latency data access
- ▶ **Elastic File System** can provide ephemeral storage to Lambda functions

## Other AWS Services

**Cognito Authentication, authorization, and user management** for web and mobile apps

**Route 53** DNS management

**Elastic IP** Re-assign static public IP addresses to your EC2 instances on-the-fly.

**Simple Notification Service (SNS)** Scalable, push-based messaging service for application-to-person or application-to-application communication. Amazon SNS enables you to send messages or notifications directly to users with SMS text messages to over 200 countries, mobile push on Apple, Android, and other platforms or email (SMTP).

## Other AWS Services (2)

An increasing number of **Machine Learning**-based services...

- ▶ Lex (chatbots)
- ▶ Polly (synthetic speech)
- ▶ Translate
- ▶ Rekognition (object and facial image recognition)
- ▶ ...

# Estimating Costs

- ▶ AWS provides a “cost calculator”
- ▶ <https://calculator.aws/>