

Hands-on Cloud Computing Services

Lezione 5

Gabriele Russo Russo
University of Rome Tor Vergata, Italy

A.A. 2022/23



TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

Exercise: Photogallery + DynamoDB

- ▶ Solution: `photogallery_v3`

AWS Elastic Beanstalk

- ▶ Platform-as-a-Service offering by AWS
- ▶ Deploy your apps on EC2 without manual infrastructure setup
- ▶ Many platforms supported
 - ▶ Java, nodeJS, Python, Go, Docker, ...
- ▶ Pricing: no additional costs besides EC2, S3 and any DB you use
- ▶ Effettuiamo il deploy di una app di esempio: [go-beanstalk.zip](#)

- ▶ Function-as-a-Service offering by AWS
- ▶ Enables the execution of serverless functions
- ▶ Functions can be written using many different languages
- ▶ Differences w.r.t. Beanstalk?
 - ▶ Fast scaling from zero to “infinity”
 - ▶ 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

Invocation example

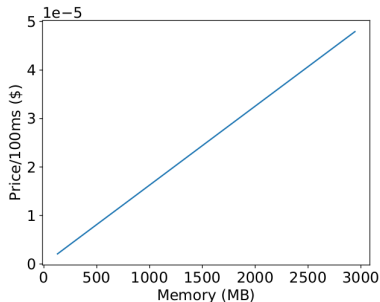
Synchronous invocation using the CLI

```
aws lambda invoke --function-name prova response.json  
# with args:  
aws lambda invoke --function-name prova  
  --payload '{ "key1": "A", "key2": "b", "key3": "c" }'  
  --cli-binary-format raw-in-base64-out  
  response.json
```

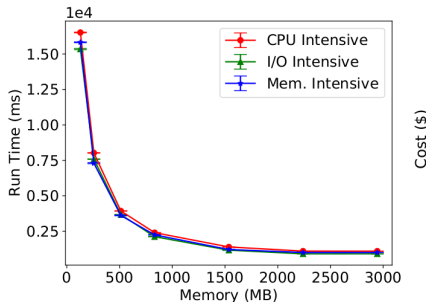
- ▶ Add `-log-type Tail` to get function log (in base64)
- ▶ Add `-invocation-type Event` for async requests¹
- ▶ Using the SDK: `invoke_hello_world.py`

¹<https://docs.aws.amazon.com/lambda/latest/dg/invocation-async.html>

AWS Lambda: Sizing



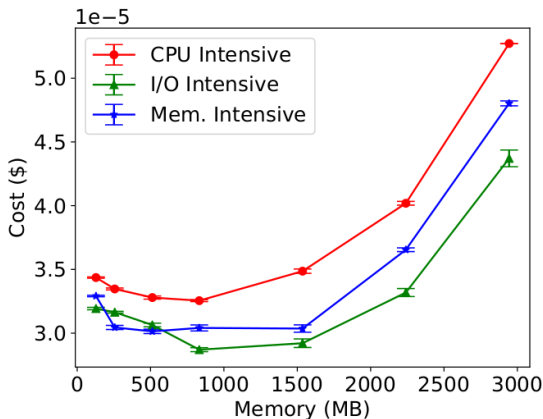
(a) AWS price per 100ms



(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

Simple Experiment

- ▶ Let's deploy a function to check if a number is prime
- ▶ We implement the function in Golang and deploy it using Terraform
- ▶ We check its duration varying the memory allocation (e.g., 128, 256 and 1024 MB)

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`

Example: Lambda + SQS + S3

- ▶ We want to integrate a Lambda function with SQS and S3
- ▶ Function invoked every time a message is available in the queue
- ▶ Function output sent to S3
- ▶ Example: `lambda/sqss3/`
- ▶ We will use Terraform to create the required components

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)

- ▶ Execute Lambda functions at the Edge in response to CloudFront events
- ▶ Customize delivered content using a function
- ▶ <https://aws.amazon.com/it/lambda/edge/>

Lambda: Versions and Aliases

- ▶ You can define multiple **versions** of a Lambda function
- ▶ Possibly, **aliases** can be associated with a version (e.g., production, testing, devel)
- ▶ You can also let AWS automatically route users to different versions (e.g., canary testing)