
Como usar o AWS Lambda como backend do Active Job

Matheus Santana

Frevo on Rails, 29 de Fevereiro de 2024

\$ whoami



Mestre em Ciência da Computação pelo Centro de
Informática da Universidade Federal de Pernambuco
(CIn/UFPE)

Desenvolvedor de aplicações Rails desde 2012

Staff Software Engineer na KnowBe4

@edmatheus no Telegram

KnowBe4
Human error. Conquered.

Estamos contratando!

Snr. Software Engineer - Ruby on Rails

Graphic Designer

Graphic Designer - Service Center

<https://www.knowbe4.com/careers/job-openings?filter=S%C3%A3o%20Paulo,%20Brazil>



Agenda

Motivação

Contexto

- Active Job
- AWS Lambda

Mão na massa

- Lamb
 - Goat
-

Motivação

Motivação

Economizar com licença do Sidekiq



Motivação

Explorar a arquitetura alternativa ao Sidekiq (SQS + Lambda)



Motivação

Encontrar as pessoas mais legais em linha reta do mundo

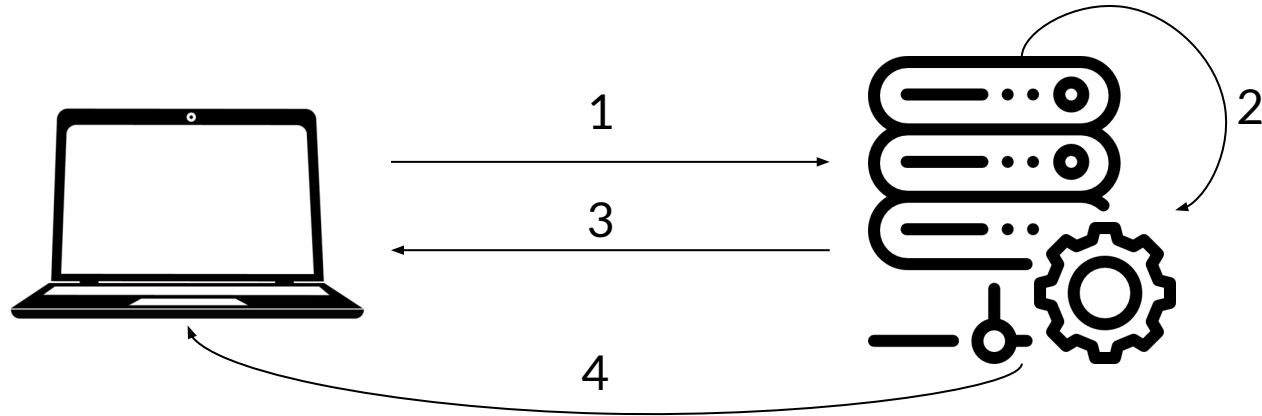


Contexto

ActiveJob - O que é?

Framework para **definição de jobs** e execução em uma **variedade de backends** de enfileiramento. Os jobs podem ser **qualquer coisa** desde limpezas programadas para rodar frequentemente a cobrança de pagamentos e envio de e-mails – qualquer coisa que possa ser fragmentada em **pequenas unidades de trabalho** que são **executadas em paralelo**.

ActiveJob - Caso de uso básico



1. Cliente envia uma requisição
 2. Servidor inicia tarefa em segundo plano
 3. Servidor responde ao cliente sem precisar aguardar a finalização da tarefa
 4. Cliente eventualmente recebe efeito da execução da tarefa
-

ActiveJob - Backends

Que

Backburner

Sucker Punch

Resque

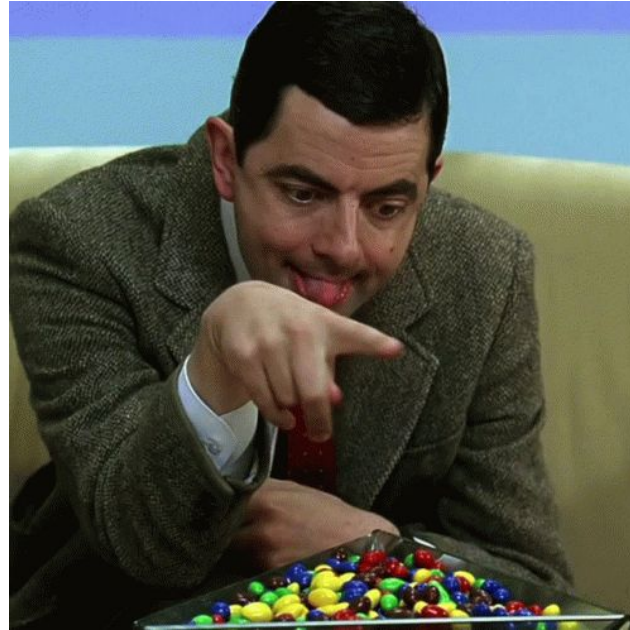
queue_classic

Sidekiq

Delayed Job

Sneakers

ActiveJob - Features dos backends



ActiveJob - Features dos backends

Async: não bloqueia a execução

Queues: execução de tarefas em diferentes *filas*

Delayed: a execução pode ser postergada

Priorities: a execução de uma tarefa ou grupo de tarefas (e.g., fila) pode ser priorizada

Timeout: interrupção da execução de uma tarefa ou grupo de tarefas depois de um tempo determinado

Retries: tentar novamente execução que falhar

ActiveJob - Features dos backends

Backend	Async	Queues	Delayed	Priorities	Timeout	Retries
Delayed Job	✓	✓	✓	Job	Global	Global
Resque	✓	✓	✓ (Gem)	Queue	Global	✓
Sidekiq	✓	✓	✓	Queue	×	Job
Sucker Punch	✓	✓	✓	×	×	×
Active Job Async	✓	✓	✓	×	×	×
Active Job Inline / Test	×	✓	N/A	N/A	N/A	N/A

AWS Lambda - O que é?

Serviço de computação que permite **executar código sem provisionar ou gerenciar servidores.**

AWS Lambda - O que é?

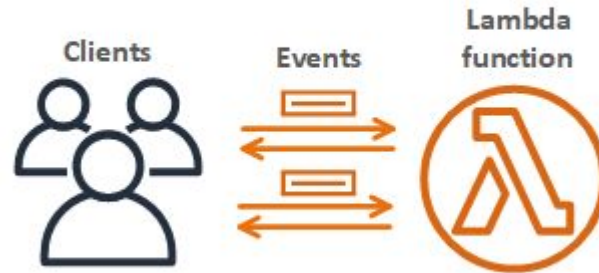
Tudo que você precisa fazer é **fornecer seu código** em um dos ambientes de execução que o Lambda suporta.

AWS Lambda - Modos de invocação



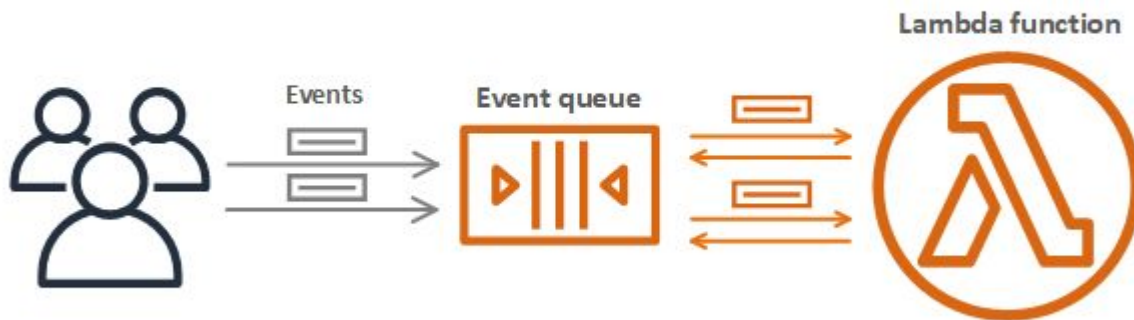
AWS Lambda - Modos de invocação

Synchronous Invocation



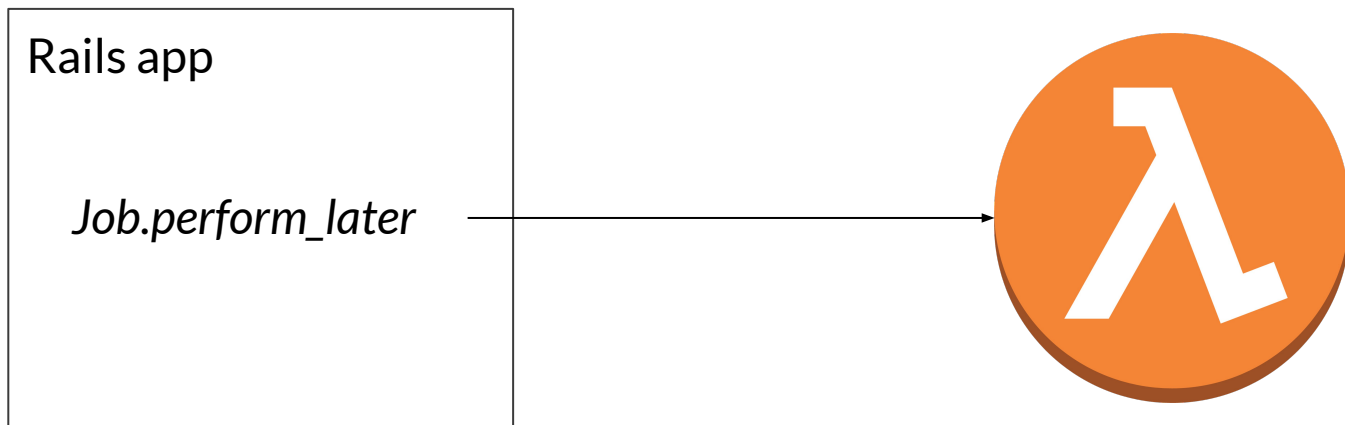
AWS Lambda - Modos de invocação

Asynchronous Invocation

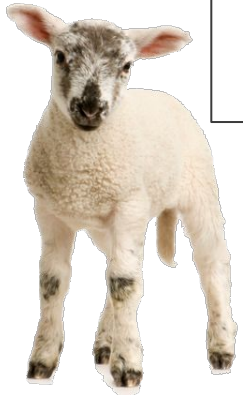
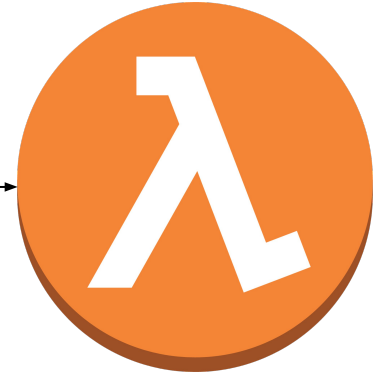
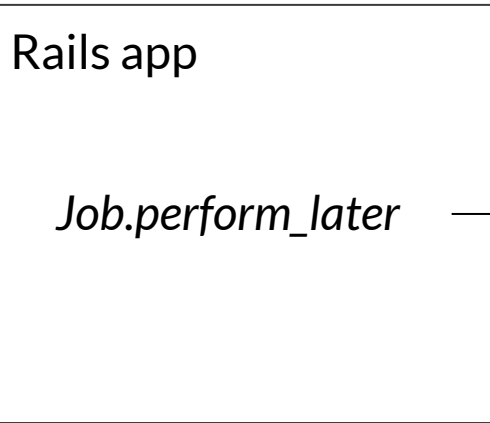


Mão na massa

Backend super básico



Backend super básico => Lamb



Lamb adapter - Gemfile

```
# Gemfile  
gem 'aws-sdk-lambda'
```



Lamb adapter - lambda_adapter.rb

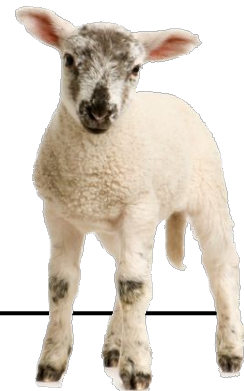
```
# lib/active_job/queue_adapters/lamb_adapter.rb
module ActiveSupport
  module QueueAdapters
    class LambAdapter
      def enqueue(job)
        client = Aws::Lambda::Client.new(region: 'us-east-1')
```



Lamb adapter - lambda_adapter.rb

```
# lib/active_job/queue_adapters/lamb_adapter.rb (continuação)
  response = client.invoke(function_name: 'lamb',
                           invocation_type: 'Event',
                           payload: JSON[job.serialize])

  puts response
end
end
end
end
```



Lamb adapter - config/application.rb

```
config.active_job.queue_adapter = :lamb
```



Lamb handler - Gemfile

```
# Gemfile  
gem 'aws_lambda_ric'
```



Lamb handler - Dockerfile

```
ENTRYPOINT [  
"/usr/local/bundle/ruby/2.7.0/gems/aws_lambda_ric-2.0.0/bin/aws_lambda_  
ric"  
]  
CMD ["config/environment.ActiveJob::LambHandler.handle"]
```



Lamb handler - Handler

```
# lib/active_job/lamb_handler.rb
module ActiveJob
  class LambdaHandler
    def self.handle(event:, context:)
      Base.execute event
    end
  end
end
```



Lamb - Recursos AWS

Identity and Access Management (IAM)

- Role e policies com permissões para o CloudWatch

Elastic Container Registry (ECR)

- Repositório

Lambda

- Função
-



Lamb - Demo

```
class WorkerJob < ApplicationJob
  def perform
    puts "I'm done working"
  end
end
```



Lamb - Features

Backend	Async	Queues	Delayed	Priorities	Timeout	Retries
Lamb	?	?	?	?	?	?



Lamb - Features

Backend	Async	Queues	Delayed	Priorities	Timeout	Retries
Lamb	✓	?	?	?	?	?



Lamb - Features

Backend	Async	Queues	Delayed	Priorities	Timeout	Retries
Lamb	✓	×	?	?	?	?



Lamb - Features

Backend	Async	Queues	Delayed	Priorities	Timeout	Retries
Lamb	✓	×	×	?	?	?



Lamb - Features

Backend	Async	Queues	Delayed	Priorities	Timeout	Retries
Lamb	✓	×	×	×	?	?



Lamb - Features

Backend	Async	Queues	Delayed	Priorities	Timeout	Retries
Lamb	✓	×	×	×	✓	?

O valor padrão de timeout de uma lambda function é de 3 segundos mas pode ser ajustado em incrementos de 1 segundo até o máximo de 15 minutos.



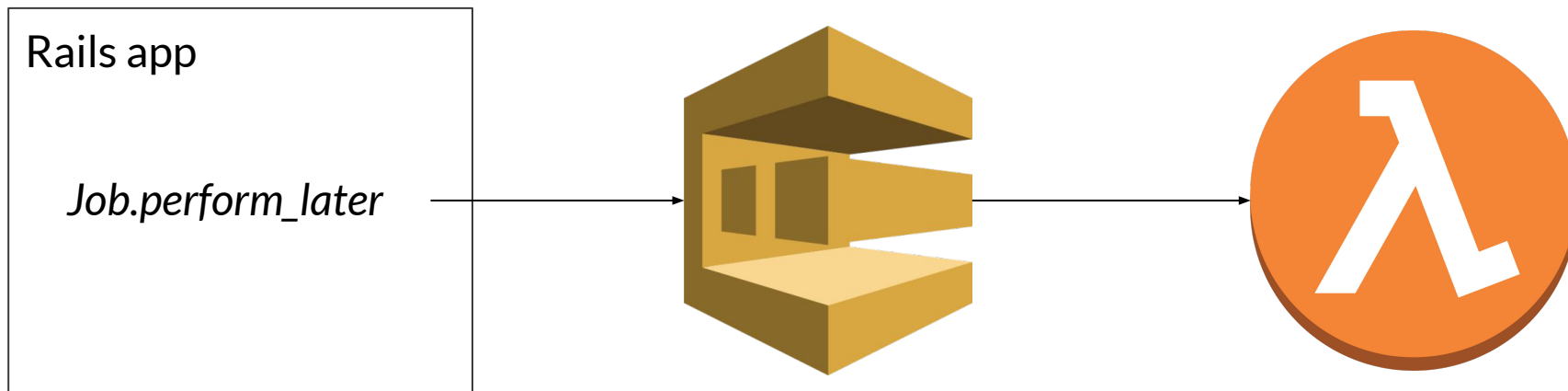
Lamb - Features

Backend	Async	Queues	Delayed	Priorities	Timeout	Retries
Lamb	✓	×	×	×	✓	✓

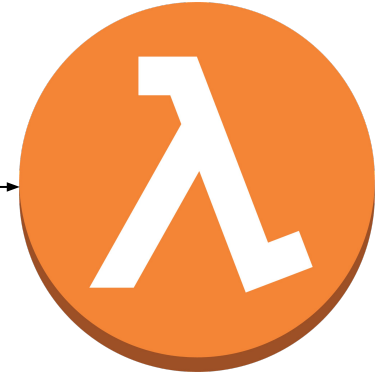
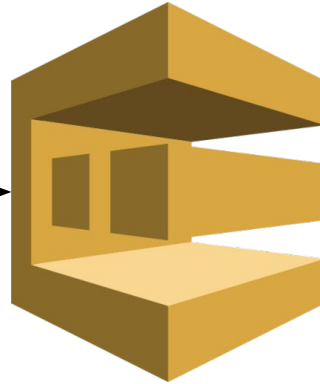
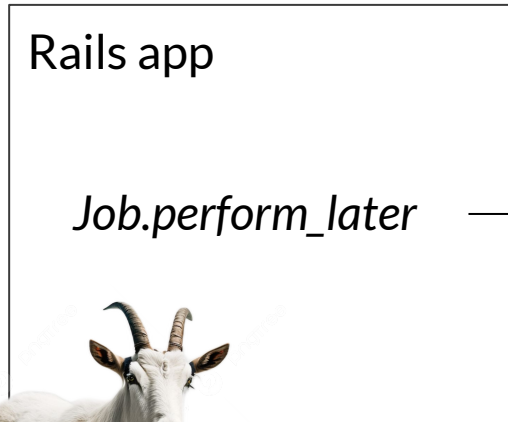
Se a função retornar um erro, ela será executada novamente mais duas vezes com 1 minuto de intervalo entre a 1ª e a 2ª tentativas e 2 minutos de intervalo entre a 2ª e a 3ª tentativas.



Lambda + SQS backend



Lambda + SQS backend => Goat



Goat - Gemfile

```
# Gemfile  
gem 'aws-sdk-rails'
```



Goat - config/application.rb

```
config.active_job.queue_adapter = :amazon
```



Goat - Dockerfile

```
CMD ["config/environment.Aws::Rails::SqsActiveJob.lambda_job_handler"]
```



Goat - Recursos AWS

Identity and Access Management (IAM)

- Role e políticas com permissões para o SQS

Simple Queue Service (SQS)

- Fila

Lambda

- Trigger (Event Source Mapping)
-



Goat - Demo



Goat - Features

Backend	Async	Queues	Delayed	Priorities	Timeout	Retries
Goat	✓	?	?	?	✓	✓



Goat - Features

Backend	Async	Queues	Delayed	Priorities	Timeout	Retries
Goat	✓	✓	?	?	✓	✓



Goat - Features

Backend	Async	Queues	Delayed	Priorities	Timeout	Retries
Goat	✓	✓	✓	?	✓	✓

Fila (delay queue) e job (message timer). O delay padrão é de 0 segundos e o máximo é de 15 minutos.



Goat - Features

Backend	Async	Queues	Delayed	Priorities	Timeout	Retries
Goat	✓	✓	✓	✓	✓	✓



Obrigado!

Dúvidas? Reclamações? Sugestões? Elogios?



Referências

Repositórios no GitHub:

- <https://github.com/embs/lamb>
 - <https://github.com/embs/goat>
 -
-

Referências

AWS:

- <https://docs.aws.amazon.com/lambda/latest/dg/welcome.html>
 - <https://docs.aws.amazon.com/lambda/latest/dg/ruby-image.html>
 - <https://docs.aws.amazon.com/lambda/latest/dg/with-sqs.html>
 - <https://docs.aws.amazon.com/sdk-for-ruby/v3/developer-guide/lambda-ruby-example-run-function.html>
-

Referências

Gems:

- <https://github.com/aws/aws-sdk-rails>
 - <https://github.com/aws/aws-lambda-ruby-runtime-interface-client>
 - <https://github.com/rails-lambda/lambdakiq>
 - <https://github.com/rails-lambda/lamby>
 -
-

Referências

Rails Docs:

- <https://api.rubyonrails.org/classes/ActiveJob.html>
 -
-

Referências

Terraform:

- [https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/lambda function](https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/lambda_function)
 - [https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/lambda event source mapping](https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/lambda_event_source_mapping)
 -
-

Referências

Misc:

- <https://rip.hibariya.org/post/creating-an-active-job-adap-ter/>
 -
-