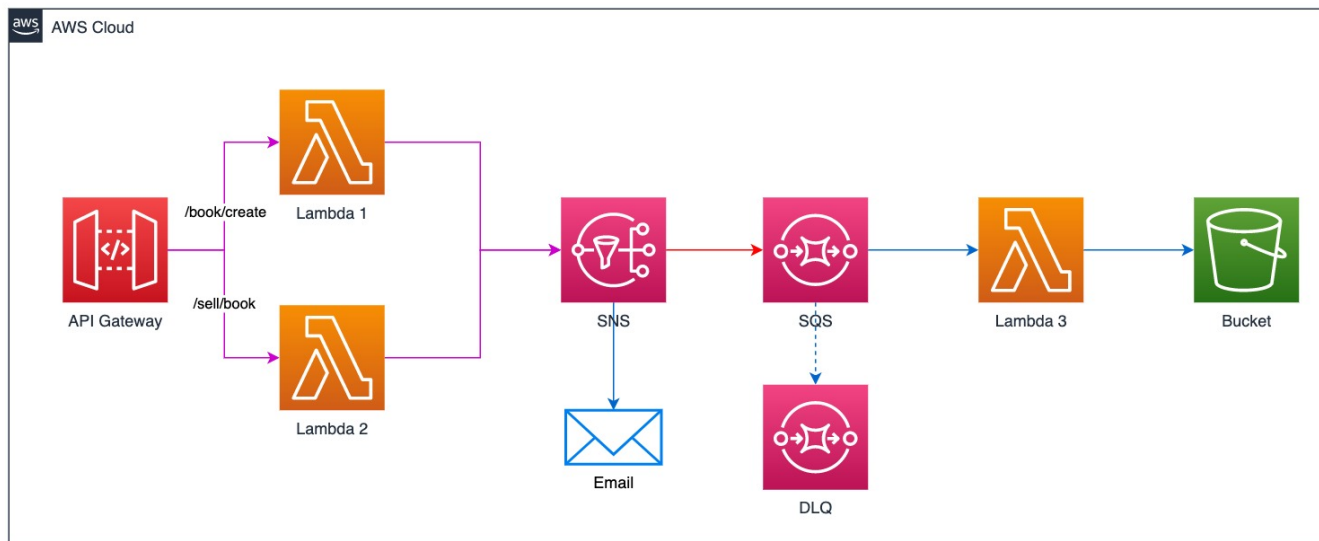


Desafio Final



O Desafio consiste em montar a arquitetura proposta acima utilizando serverless framework e terraform.

Em Terraform:

1. SNS
2. SQS
3. SQS DLQ
4. Subscrição do seu email ao SNS
5. SUBscrição do SQS ao SNS
6. Política necessária no SQS para o Item 5

Em Serverless Framework

1. API Gateway
2. Lambda1
3. Lambda2
4. Lambda3
5. Integração do API Gateway aos lambdas 1 e 2
6. Gatilho vindo do SQS ao Lambda3

Linguagem de programação

Pode ser utilizada qualquer linguagem de programação disponível para o lambda. [FAQ Lambda](#)

Caso escolha Python, existem varios exemplos e códigos prontos apresentados em aula.

Entrega

Um zip do projeto contendo uma pasta para os arquivos do Terraform e outra para o Serverless Framework com os seguintes conteúdos:

1. Terraform:
 1. Arquivos .tf para criação dos recursos propostos
 2. Um arquivo para guarda o estado remotamente
2. Serverless Framework
 1. Serverless.yml unico
 2. Código para todos os lambdas
3. Script ou manual de como fazer o deploy em qualquer conta AWS

Regras

1. Maximo de recebimentos da fila principal antes de mandar para DLQ = 2.
2. Request do endpoint /book/create deve seguir o exemplo:

```
{
  "book_name":"harry potter",
  "book_id": 34577,
  "book_preco": 45.87
}
```

3. Request do endpoint /sell/book seguir o exemplo:

```
{
  "book_id":2346,
  "customer_id": 12456
}
```

4. A resposta das APIs podem ser como desejarem desde que responda com 200.
5. A subscrição tanto do email quanto sqs no tópico SNS serão feitos via Terraform.
6. Cada requisição deve virar um arquivo no s3.

Dicas

1. [Resource: aws_sqs_queue](#)
2. [sqs-with-dlq](#)
3. [API Gateway no serverless](#)
4. [SQS trigger Lambda](#)
5. [Resource: aws_sns_topic_subscription](#)
6. [Resource: aws_sns_topic](#)
7. [SNS para SQS subscription](#)
8. [Exemplos extras de códigos](#)
9. Não se esqueça que os lambdas precisam de autorização para acessar os serviços e que isso deve ser configurado no serverless.yml
10. Em caso de erro verifique o log do lambda. [Lambda-Logs-serverless-reference](#) [Lambda-Logs-Cloudwatch-reference](#)