

Curso: Spring Boot com Ionic - Estudo de Caso Completo

Capítulo: Armazenamento de imagens usando Amazon S3

Objetivo geral:

- ⌚ Armazenar arquivos usando o serviço S3 da Amazon AWS ⌚
Salvar imagens de perfil dos clientes
- ⌚ Bônus: manipular imagens para converter formato, "cropar" e redimensionar

Visão geral

Sobre o free tier da AWS:

<https://aws.amazon.com/pt/free/>

Sobre S3:

<https://aws.amazon.com/pt/s3/>

Preços:

<https://aws.amazon.com/pt/s3/pricing/>

Post comparativo de preços:

<https://www.msp360.com/resources/blog/amazon-s3-azure-and-google-cloud-prices-compare>

Criação de uma conta da AWS

<https://aws.amazon.com/s3/>

Contrato do cliente AWS:

<https://aws.amazon.com/pt/agreement/>

Criando um bucket no S3

- ⌚ No console AWS, acessar S3
- ⌚ Create Bucket -> dê um nome e selecione a região

Setup do IAM - Identity Access Management

IAM é o serviço central que gerencia a segurança da conta AWS.

Acessar o dashboard -> Security, Identity & Compliance -> IAM

Setup do MFA - Multi-factor authentication (OPCIONAL)

- 🕒 Instalar o Google Authenticator App no seu smartphone
- 🕒 Clicar no botão "Manage MFA" e selecione "A virtual MFA device" 🕒
- Ler o QR Code a partir do app Google Authenticator
- 🕒 Entrar com dois códigos gerados pelo app e clique em "Activate Virtual MFA"

Create individual IAM users

- 🕒 Manage users -> Create new users
- 🕒 Crie um usuário para seu sistema acessar o S3 (exemplo: "curso_spring_user")
- 🕒 Baixe o arquivo com as credenciais do usuário (user name, access key id, secret access key)

Use groups to assign permissions

- 🕒 Manage groups -> Create new group
 - 🕒 Criar um grupo (exemplo: "developers")
 - 🕒 Busque por "S3" e selecione "AmazonS3FullAccess" e confirme
 - 🕒 Selecione o grupo e clique: Group Actions -> Add Users do Group 🕒
- Selecione o usuário e confirme

Apply an IAM password policy (OPCIONAL)

- 🕒 Manage Password Policy
- 🕒 Selecione as políticas desejadas

Salvando primeiro arquivo no S3

Checklist:

1) Incluir dependências:

```
<dependency>
  <groupId>com.amazonaws</groupId>
  <artifactId>aws-java-sdk</artifactId>
  <version>LATEST</version>
</dependency>

<dependency>
  <groupId>commons-io</groupId>
  <artifactId>commons-io</artifactId>
  <version>LATEST</version>
</dependency>
```

2) Acrescentar em application.properties:

```
aws.access_key_id=
aws.secret_access_key=
s3.bucket=curso-spring-ionic
s3.region=sa-east-1
```

Veja: <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html>

- 3) Criar arquivo de configuração S3Config com um bean do tipo AmazonS3
- 4) Criar o serviço S3Service
- 5) No programa principal, fazer um teste com um arquivo do seu sistema de arquivos

Tornando o bucket com acesso público para leitura

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "AllowPublicRead",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::curso-spring-ionic/*"
      ]
    }
  ]
}
```

Enviando imagem via endpoint

Checklist:

- ⌚ Retirar código de teste no programa principal ⌚
- Atualizar o S3Service
- ⌚ Criar um método em ClienteService
- ⌚ Criar o endpoint /clientes/picture em ClienteResource
- ⌚ Em SecurityConfig, liberar acesso POST a /clientes e também provisoriamente o acesso a /clientes/picture

Tratando exceções adequadamente

Checklist:

- ⌚ Criar uma exceção personalizada de serviço FileNotFoundException
- ⌚ Em S3Service, trocar RuntimeException por FileNotFoundException
- ⌚ Acrescentar os tratamentos em ResourceExceptionHandler
 - o FileNotFoundException
 - o AmazonServiceException
 - o AmazonClientException
 - o AmazonS3Exception

Salvando a URL da imagem em Cliente

Checklist:

- ⌚ Em SecurityConfig, exigir autorização para /clientes/picture ⌚
- Em Cliente, incluir o atributo imageUrl
- ⌚ Em ClienteService, fazer as devidas alterações
- ⌚ Sugestão: revisar o tempo de expiração do token

Usando padrão de nomes para imagens

Checklist:

- ⌚ Em Cliente, remover o atributo imageUrl
- ⌚ Em application.properties, incluir: img.prefix.client.profile=cp
- ⌚ Crie um serviço ImageService com uma função para obter uma imagem JPG a partir do arquivo
 - o public BufferedImage getJpgImageFromFile(MultipartFile uploadedFile) ⌚
- Em ClienteService, fazer as devidas alterações

Bônus: ajustando tamanho da imagem

Checklist:

- ⌚ Em application.properties, incluir: img.profile.size=200
- ⌚ Incluir a dependência:

```
<dependency>
  <groupId>org.imgscalr</groupId>
  <artifactId>imgscalr-lib</artifactId>
  <version>4.2</version>
</dependency>
```

- ⌚ Em ImageService, incluir uma função para "cortar" uma imagem para que fique quadrada ⌚
- Em ImageService, incluir uma função para redimensionar uma imagem
- ⌚ Em ClienteService, fazer as devidas alterações