03

Listando Buckets via SDK

Transcrição

Vimos a necessidade do uso do SDK para trabalharmos de maneira programática na comunicação com o S3, então, precisaremos escolher qual SDK utilizaremos.

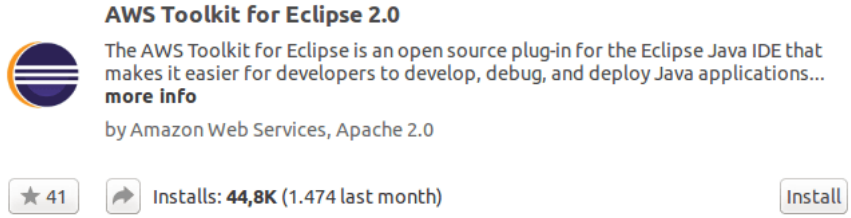
Escolhendo o SDK

No menu lateral da esquerda do site da Amazon (https://aws.amazon.com/pt/), em "Desenvolvedores -> Ferramentas", ou neste link (https://aws.amazon.com/pt/tools/), há um local específico dos SDKs (https://aws.amazon.com/pt/tools/#sdk), para escolhermos aquele que seja o mais apropriado para a nossa necessidade, isto é, para a linguagem em que estamos desenvolvendo. Independentemente de qual seja, a abordagem será praticamente a mesma, mudando somente as características de cada linguagem.

No nosso exemplo, utilizaremos o **Java**, cuja documentação (http://docs.aws.amazon.com/pt_br/sdk-for-java/v1/developer-guide/welcome.html) poderá ser consultada para entendermos como utilizaremos o seu SDK. Neste caso, já existe um *plugin* para o Eclipse, que instalaremos, e o qual basicamente baixará todas as dependências da Amazon para nós.

Criando um projeto Java AWS

Então, abriremos o Eclipse e instalaremos o *plugin*, acessando o menu superior e "Help -> Eclipse Marketplace...". Na barra de busca, pesquisaremos por "AWS", e dentre os resultados será exibido o "AWS Toolkit for Eclipse 2.0":



Clicaremos no botão "Install" e aguardaremos sua instalação, e quando o Eclipse for reinicializado, cancelaremos a solicitação de chaves caso apareça, e criaremos um **AWS Java Project** com nome `alura-s3`. No momento da criação do projeto, há o campo "AWS Credentials", as quais se associam às chaves configuradas no **AWS CLI**. Vamos manter o padrão, e então aguardaremos a criação do projeto com todas as dependências necessários para trabalharmos com o SDK da Amazon.

O projeto utiliza o Maven, uma ferramenta de *build* que lida com as dependências necessárias para trabalharmos com o SDK. Em `src/main/java` existe um pacote já criado, em que colocaremos a classe `S3Exemplo` com o método `main`:

```
public class S3Exemplo {

    public static void main(String[] args) {

    }

}
```

Vamos trabalhar com o básico da API, algo similar em qualquer outra linguagem utilizada.

Autenticação via SDK

O primeiro passo consiste em acessarmos o S3, mas antes disso precisaremos ter acesso ao AWS e, por sua vez, para isso são necessárias as credenciais, ***Access Key ID*** e ***Secret Access Key***. Podemos pegar essas informações da nossa conta do AWS, mas como já configuramos as credenciais na nossa máquina para utilizar o **AWS CLI**, vamos pegá-las no arquivo `credentials` do diretório oculto `.aws`, na pasta do nosso usuário.

Assim, vamos criar duas variáveis para guardarmos os seus valores:

```
public class S3Exemplo {

    public static void main(String[] args) {

        String accessKey = "Digite aqui sua Access Key ID";
        String secretKey = "Digite aqui dua Secret Access Key";
    }

}
```

Lembrando que nunca devemos deixar essas informações públicas, pois elas dão total acesso à nossa conta do AWS. Agora, com o SDK no nosso projeto, vamos instanciar um `BasicAWSCredentials`, passando as credenciais para o seu construtor:

```
public class S3Exemplo {

    public static void main(String[] args) {

        String accessKey = "Digite aqui sua Access Key ID";
        String secretKey = "Digite aqui dua Secret Access Key";

        BasicAWSCredentials awsCredentials = new BasicAWSCredentials(accessKey,
                secretKey);
    }

}
```

Com isso, acessaremos o S3, e para isto existe um *builder*, o `AmazonS3ClientBuilder`, que possui o método `standard` que mantém o S3 com as configurações padrões. Sendo assim, precisaremos configurá-lo de acordo com a nossa necessidade.

Nas credenciais, chamaremos o método `withCredentials` e, como as colocamos em nosso código, precisaremos provê-las por meio de `AWSStaticCredentialsProvider`:

```
public class S3Exemplo {

    public static void main(String[] args) {

        String accessKey = "Digite aqui sua Access Key ID";
        String secretKey = "Digite aqui dua Secret Access Key";
```

```
        BasicAWSCredentials awsCredentials = new BasicAWSCredentials(accessKey,
                secretKey);

        AmazonS3ClientBuilder.standard().withCredentials(
                new AWSStaticCredentialsProvider(awsCredentials));
    }

}
```

Então, criaremos um *builder* com as configurações padrões, para o qual passaremos as credenciais de acesso. Além disso, é possível fazermos outras configurações, como a região:

```
public class S3Exemplo {

    public static void main(String[] args) {

        String accessKey = "Digite aqui sua Access Key ID";
        String secretKey = "Digite aqui dua Secret Access Key";

        BasicAWSCredentials awsCredentials = new BasicAWSCredentials(accessKey,
                secretKey);

        AmazonS3ClientBuilder
                .standard()
                .withCredentials(
                        new AWSStaticCredentialsProvider(awsCredentials))
                .withRegion(Regions.SA_EAST_1);
    }

}
```

Configuradas as credenciais e a região, chamaremos o método `build` e atribuiremos o retorno a uma variável:

```
public class S3Exemplo {

    public static void main(String[] args) {

        String accessKey = "Digite aqui sua Access Key ID";
        String secretKey = "Digite aqui dua Secret Access Key";

        BasicAWSCredentials awsCredentials = new BasicAWSCredentials(accessKey,
                secretKey);

        AmazonS3 s3 = AmazonS3ClientBuilder
                .standard()
                .withCredentials(
                        new AWSStaticCredentialsProvider(awsCredentials))
                .withRegion(Regions.SA_EAST_1).build();
    }

}
```

Desta forma, temos acesso ao S3 e poderemos fazer o que quisermos com ele, de maneira programática.

Listando *buckets* via SDK

Por exemplo, poderemos listar os *buckets* por meio do método `listBuckets()`. Esse método nos retorna uma lista de *buckets* para podermos iterar sobre ela e imprimir cada *bucket* do nosso S3:

```
public class S3Exemplo {

    public static void main(String[] args) {

        String accessKey = "Digite aqui sua Access Key ID";
        String secretKey = "Digite aqui dua Secret Access Key";

        BasicAWSCredentials awsCredentials = new BasicAWSCredentials(accessKey,
                secretKey);

        AmazonS3 s3 = AmazonS3ClientBuilder
                .standard()
                .withCredentials(
                        new AWSStaticCredentialsProvider(awsCredentials))
                .withRegion(Regions.SA_EAST_1).build();

        System.out.println("Listando buckets...");
        List<Bucket> listBuckets = s3.listBuckets();
        for (Bucket bucket : listBuckets) {
            System.out.println(bucket);
        }
    }

}
```

Ao executarmos a classe, veremos os *buckets* listados no console, de maneira que conseguimos lidar com o S3 programaticamente. Por enquanto estamos listando *buckets*, mas conseguiremos fazer outras operações, que veremos mais adiante.