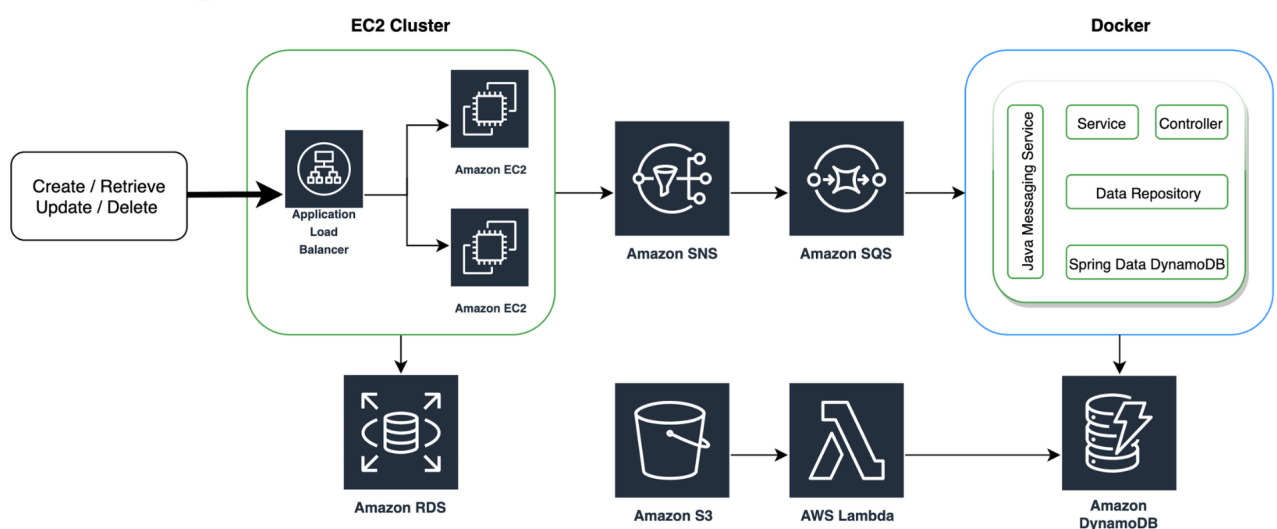




siecola.com.br

# Desenvolvendo aplicações em Java para AWS



PAULO SIECOLA

# Desenvolvendo aplicações em Java para AWS

Crie aplicações Java com Spring Boot e Docker na AWS  
com o ECS - Elastic Container Service

Paulo Cesar Siecola

Esse livro está à venda em <http://leanpub.com/amazonwebservice>

Essa versão foi publicada em 2021-04-03



Esse é um livro [Leanpub](#). A Leanpub dá poderes aos autores e editores a partir do processo de Publicação Lean. [Publicação Lean](#) é a ação de publicar um ebook em desenvolvimento com ferramentas leves e muitas iterações para conseguir feedbacks dos leitores, pivotar até que você tenha o livro ideal e então conseguir tração.

© 2018 - 2021 Paulo Cesar Siecola



*Para Joana e Rogério, meus dedicados pais. Um agradecimento especial para Isabel Mendes, minha esposa, pela excelente revisão técnica.*

# Conteúdo

|   |           |
|---|-----------|
| <b>1 - Introdução</b>                               | <b>1</b>  |
| 1.1 - A quem se destina esse livro                  | 1         |
| 1.2 - O que é AWS                                   | 1         |
| 1.3 - Serviços e recursos da AWS                    | 2         |
| 1.4 - Estrutura didática do livro                   | 8         |
| 1.5 - O que será necessário                         | 8         |
| 1.6 - Limitações de recursos AWS pelo tipo de conta | 9         |
| 1.7 - Próximos passos                               | 9         |
| <b>2 - Preparação do ambiente</b>                   | <b>10</b> |
| 2.1 - Console da AWS                                | 10        |
| 2.2 - Ambiente de desenvolvimento                   | 12        |
| 2.3 ) Docker  | 13        |
| 2.4 - Postman                                       | 14        |
| 2.5 - Conclusão                                     | 15        |
| <b>3 - Conteúdo do livro</b>                        | <b>16</b> |



# 1 - Introdução

Bem vindo ao livro **Desenvolvendo aplicações em Java para AWS**! Nele será ensinado como desenvolver aplicações que serão executadas no ambiente Amazon Web Services. Além disso, o leitor entenderá como funcionam diversos serviços de *cloud computing* e suas interações através de aplicações que serão construídas ao longo dos capítulos.

O leitor que decidiu seguir esse livro já deve saber da relevância que o desenvolvimento de aplicações utilizando *cloud computing* tem nas diversas áreas de tecnologia e negócios. Muito mais do que uma tendência, esse tipo de abordagem, para criação de sistemas e aplicações de software, tornou-se algo necessário para várias áreas de negócio. Atualmente poucos segmentos ainda utilizam infraestrutura própria para hospedagem de suas aplicações, talvez ainda motivados por insegurança e desconhecimento na infraestrutura de *cloud computing*.

Para acompanhar esse livro, é necessário que o leitor tenha conhecimento prévio de desenvolvimento de aplicações utilizando a linguagem **Java e programação orientada a objetos**. Conhecimentos específicos da linguagem, tais como *frameworks* e ferramentas de desenvolvimento serão apresentados ao longo do livro.

As demais linguagens que aparecerem serão utilizadas minimamente para construção de *scripts* ou arquivos de configuração, necessários para construção das aplicações e configuração dos serviços que serão criados.

As aplicações ensinadas nesse livro, como forma de exemplificação dos conceitos que serão apresentados, utilizarão *frameworks* e ferramentas modernas, como por exemplo **Spring Boot** e **Docker**. Dessa forma, o leitor aprenderá a trabalhar com tais tecnologias em conjunto com os serviços de *cloud computing*.

## 1.1 - A quem se destina esse livro

Esse livro é destinado a **desenvolvedores de software**, com ou sem experiência em *cloud computing* e desenvolvimento Web, que desejam se aprofundar nos serviços oferecidos pelo Amazon Web Services.

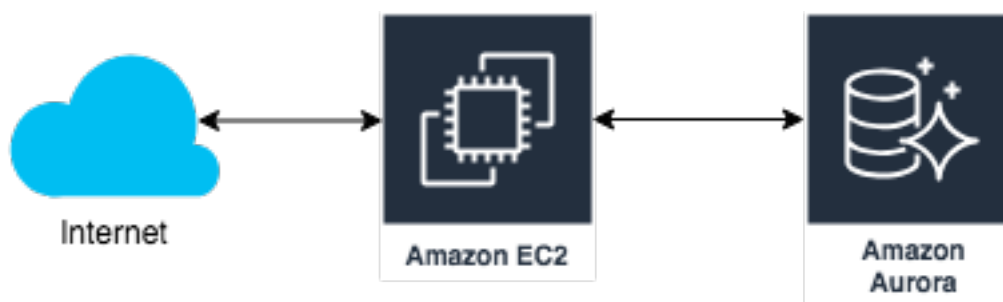
Também pode servir como uma fonte de conhecimento para **administradores e operadores de sistemas**, como forma de se familiarizarem com as ferramentas de criação e administração de recursos de *cloud computing* da Amazon Web Services.

## 1.2 - O que é AWS

AWS é a sigla para Amazon Web Service, a plataforma de *cloud computing* da Amazon. Nela é possível criar aplicações para diversas finalidades, hospedadas numa infraestrutura robusta e

escalável, utilizando serviços como banco de dados, servidores, dispositivos de armazenamento de dados e muito mais.

Um exemplo típico do que se pode fazer utilizando AWS, é criar uma máquina virtual para executar uma aplicação e conectá-la a um banco de dados para armazenamento de informações, como mostra a figura a seguir:



Exemplo de aplicação com banco de dados

Os dois recursos mostrados na figura são serviços da AWS e serão detalhados a seguir, mas eles representam uma instância de uma máquina virtual (Amazon EC2), onde a aplicação é executada, e um serviço para hospedagem de banco de dados (Aurora), para armazenamento das informações da aplicação.

Até esse ponto não há nada de novo, porém esses recursos (a máquina virtual e o servidor do banco de dados), podem ser ligados ou desligados quando necessário, além de serem alterados em função de processamento e armazenamento à medida que a demanda da aplicação cresce ou diminui. Com isso, o custo de utilização de tais recursos pode ser ajustado de acordo com o número de requisições que a aplicação recebe.

Essa característica de utilização sob demanda de recursos de *cloud computing* é uma das grandes vantagens dessa tecnologia. Os custos de utilização dos recursos e serviços acompanham, de certa forma, a variação de sua utilização, o que torna seu uso interessante do ponto de vista de negócios e financeiramente mais viável.

Os recursos podem ser acessados diretamente da Internet ou protegidos com **regras de segurança** baseadas em diversos parâmetros de configuração, como por exemplo somente serem acessíveis através de outros recursos da AWS, ou seja, totalmente bloqueados do “mundo externo”.



Segurança de acesso a recursos e informações é algo fundamental em aplicações de *cloud computing*, e uma característica fortemente presente nos serviços oferecidos pela AWS.

## 1.3 - Serviços e recursos da AWS

Nessa sessão serão apresentados alguns dos **serviços e recursos** oferecidos pela AWS, principalmente os que serão abordados nesse livro.

O intuito é apresentar brevemente alguns serviços com suas nomenclaturas e símbolos, que serão utilizados nos diagramas apresentados ao longo do livro, para que o leitor se acostume quando eles aparecerem.

Todos os recursos apresentados aqui serão abordados posteriormente no livro, ao longo de seus capítulos. Porém, é possível que, por questões didáticas, alguns recursos sejam citados antes de serem formalmente apresentados, logo, é importante uma breve introdução a eles.

## a) Amazon Elastic Compute Cloud - EC2

Talvez esse seja o serviço mais conhecido que a AWS provê. Com ele é possível criar **instâncias de máquinas virtuais** com capacidade de processamento, memória e armazenamento variados.

Durante o processo de criação de uma instância EC2, é possível escolher o sistema operacional que irá executar na máquina virtual. Com isso, após sua criação, a máquina virtual ficará pronta para ser utilizada.

O símbolo utilizado para representar uma instância EC2 em diagramas é o seguinte:



Com uma instância EC2, é possível, por exemplo, instalar um servidor e hospedar uma aplicação Web disponível para toda a Internet. Também é possível executar uma aplicação dentro de um *container*, utilizando tecnologias como o Docker, que será utilizado e explicado ao longo dos capítulos desse livro).

## b) Amazon Elastic Container Service - ECS

Com o ECS é possível executar aplicações dentro de **containers, como o Docker**. Dessa forma, não é necessário cuidar da administração de instâncias de máquinas virtuais EC2, que hospedariam tais aplicações.

Dentro de uma ECS, o administrador do sistema pode criar tarefas para executar várias máquinas virtuais, fazendo que a mesma aplicação tenha várias instâncias em execução ao mesmo tempo. Dessa forma, pode-se fazer um balanceamento de carga de processamento e tratamento de requisições entre as várias instâncias que estão rodando a aplicação. E ainda, é possível aumentar ou diminuir a quantidade de instâncias, automaticamente, de acordo com tais parâmetros de processamento e quantidade de requisições.



O símbolo que representa um ECS é o seguinte:



Executar aplicações dentro de *containers* traz muitas vantagens, principalmente no que se refere à orquestração da quantidade de instâncias da aplicação que devem ser executadas e para onde o tráfego deve ser direcionado, em caso de grande volume de requisições ou processamento. Esse é um assunto que será tratado nesse livro no momento que o conceito de ECS for explorado com mais detalhes.

## c) Amazon CloudWatch

O CloudWatch é um serviço da Amazon Web Service de monitoramento e gerenciamento. Algumas de suas funções mais comuns são:

- Visualização de **logs** gerados por outros recursos e aplicações;
- Gráficos de **métricas** como consumo de memória e processamento;
- Visualização de **eventos**.

O símbolo que representa o CloudWatch é o seguinte:



Em um sistema com aplicações distribuídas em várias instâncias ou com diferentes tipos de recursos, é importante ter uma forma de concentrar essas informações em um único lugar, para facilitar seu monitoramento.

## d) Amazon Relational Database Service - RDS

Este é um serviço para criação de **bancos de dados relacionais** na AWS, como **Aurora, MySQL, PostgreSQL, Oracle e SQL Server**, sem a necessidade da administração e operação de servidores.

O símbolo que representa o RDS é o seguinte:



Amazon RDS

Amazon RDS

O RDS também oferece benefícios como:

- Agendamento de backups;
- Recuperação de dados;
- Réplica de dados;
- Monitoramento e métricas.

## e) Amazon DynamoDB

O Amazon DynamoDB é um banco de dados baseado em **documentos e estruturas de chave-valor**. Bem diferente de um banco de dados relacional, em termos de estrutura de armazenamento de dados, esse serviço possui operações de acesso extremamente rápidas em qualquer escala.

O símbolo que representa o Amazon DynamoDB é o seguinte:



Amazon  
DynamoDB

AWS DynamoDB

Algumas de suas características são:

- Agendamento de backups;

- Recuperação de dados;
- Mecanismo que possibilita avisar a aplicação em caso de mudança de registros;
- Alta performance para operações de escrita e leitura;
- Mecanismo para exclusão automática de registros.

## f) Amazon Simple Queue Service - SQS

O SQS é o **mecanismo de filas para troca de mensagens** da AWS, para, por exemplo, o estabelecimento de comunicação entre aplicações diferentes.

O símbolo que representa o Amazon SQS é o seguinte:



Amazon SQS

AWS SQS

A seguir, algumas das características do SQS:

- Para começar, é necessário conectar o IntelliJ IDEA ao Docker da máquina de desenvolvimento. Isso pode ser feito acessando suas configurações, no menu Preferences\Settings -> Build, Execution, Deployment, na seção Docker, como pode ser visto na figura a seguir:
- Garantia de entrega de pelo menos uma mensagem;
- Possui mecanismos para evitar inversão de ordem de mensagens.

## g) Amazon Simple Notification Service - SNS

Com o AWS SNS é possível enviar mensagens de **notificações** para dispositivos móveis, como Android e iOS, além de disparar eventos para outros recursos AWS. Ele também pode trabalhar de forma integrada com o AWS SQS, para permitir a propagação de mensagens de uma aplicação para várias outras, ao mesmo tempo.

O símbolo que representa o Amazon SNS é o seguinte:





Amazon SNS

AWS SNS

A seguir, algumas de suas características:

- Permite a integração com outros recursos AWS, como SQS, Lambda e S3;
- Vários SQS podem se registrar num SNS para recebimento das notificações;
- Independente de protocolo de mensagens.

## h) Amazon Simple Storage Service - S3

O S3 da AWS é um dos seus serviços de **armazenamento de objetos em massa**. Ele é especialmente utilizado para armazenamento de arquivos, de forma temporária ou não, com alta disponibilidade de acesso.

O símbolo que representa do AWS S3 é o seguinte:



Amazon S3

AWS S3

A seguir, algumas de suas características:

- Alta disponibilidade;
- Permite notificar outros recursos AWS quando um objeto novo é inserido;
- Possui configurações de segurança para restringir o acesso público ou privado.

## i) AWS Lambda

O Lambda é o serviço para **execução de código da AWS sem a utilização de servidores**, ou como é essa técnica conhecida: *serverless*.

Com ele é possível reagir a notificações ou eventos e executar um pequeno trecho de código de forma rápida, sem a necessidade de manter uma infraestrutura de servidores ou instâncias de máquinas virtuais ociosas.

O símbolo que o representa é o seguinte:



AWS Lambda

AWS Lambda

Algumas das principais características do AWS Lambda são:

- Não é necessário criar servidores para execução do código;
- Várias instâncias podem ser executadas para atender qualquer demanda de requisições;
- O custo de sua utilização é cobrado por tempo de execução de código, ou seja, somente quando é necessário executar algo.



Existem muitos outros recursos e serviços que são oferecidos pelo AWS. Para ter uma lista atualizada, consulte o seguinte link: <https://aws.amazon.com>

## 1.4 - Estrutura didática do livro

A estrutura didática desse livro permeia um conceito conhecido como **aprendizado baseado em problemas**, onde o leitor é apresentado e conduzido aos conceitos-chaves através de problemas que devem ser resolvidos utilizando tais conhecimentos.

Obviamente, nem todos os conceitos podem ser apresentados dessa forma, mas para aqueles que permitem, será utilizado uma abordagem mais prática de aprendizado, fazendo com que o leitor sempre tenha em mente um problema a ser resolvido utilizando a tecnologia que é detalhadamente apresentada. Dessa forma, tendo-se sempre em mente o problema-alvo a ser resolvido, o leitor pode absorver os conceitos que são apresentados de forma mais eficiente.

## 1.5 - O que será necessário

O próximo capítulo detalha tudo o que o leitor precisará para acompanhar esse livro. Aqui segue um resumo para que possa ser percebido que nada de especial ou com custo será utilizado:

- Computador com Windows, OS X ou Linux;
- Ferramentas de desenvolvimento gratuitas;
- Conta de usuário na AWS, que pode ser criada de várias formas para um período gratuito de avaliação;
- Repositório de código no GitHub.

A ferramenta Postman será amplamente utilizada nesse livro, para acessar serviços REST que serão criados. É importante que o leitor tenha familiaridade com tal ferramenta.

## 1.6 - Limitações de recursos AWS pelo tipo de conta

Existem certas restrições de recursos que a AWS impõe, dependendo do tipo de conta utilizada para acessá-la. Um exemplo disso é o programa [AWS Educate](https://aws.amazon.com/education/awseducate/)<sup>1</sup>, que oferece créditos para estudantes e professores, mas que limita alguns recursos ou parte deles.

Essas restrições estão em constante revisão pela AWS, por isso verifique as que são impostas ao tipo de conta escolhido.

## 1.7 - Próximos passos

Nesse capítulo o leitor teve uma breve introdução do que é AWS, o serviço de *cloud computing* da Amazon, e alguns dos recursos que ela oferece. Além disso, foi apresentado como o livro está estruturado.

No próximo capítulo o leitor verá o que deve fazer para preparar seu ambiente de desenvolvimento e também os primeiros passos que deverão ser dados no console de administração da AWS para que ele possa começar a criar os recursos e desenvolver as aplicações.

---

<sup>1</sup><https://aws.amazon.com/education/awseducate/>



## 2 - Preparação do ambiente

Este capítulo descreve tudo o que deverá ser instalado ou preparado para acompanhar esse livro:

- Conta na AWS;
- Ambiente de desenvolvimento com o IntelliJ IDEA
- Outras ferramentas, como o Postman, que é um cliente REST.

### 2.1 - Console da AWS

A Amazon possui um console de administração para se utilizar a AWS, para criar, gerenciar recursos e posteriormente executar aplicações.

Esse console pode ser acessado no seguinte endereço: <http://console.aws.amazon.com/>

#### a) Tipos de contas na Amazon Web Services

Para acessar a AWS, através de seu console, é necessário criar uma conta. Dessa forma, será possível controlar a utilização de recursos, gastos (quando aplicáveis), monitorar alarmes de utilização, dentre várias outras funcionalidades.

A AWS possui alguns tipos de contas descritas logo a seguir. Qualquer uma dessas contas pode ser utilizadas para o acompanhamento desse livro:

- **Free tier:** talvez essa seja a forma mais simples de começar a utilizar a AWS para fins de avaliação e aprendizado. Com uma conta desse tipo é possível ter 12 meses gratuitos para utilizar alguns dos recursos da AWS, dentro de certos limites.

Constantemente a AWS renova os recursos e os limites desse tipo de conta. Para ter uma visão atualizada sobre essa modalidade, acesse o seguinte link: <https://aws.amazon.com/free>

Essa modalidade requer o uso de um cartão de crédito para a criação da conta, mas sem cobranças caso a utilização fique dentro dos limites estabelecidos nos critérios do *free tier* da AWS.

- **AWS Educate:** essa é uma opção ideal para professores e alunos de escolas, faculdades e universidades, pois oferece várias modalidades de créditos mensais para os usuários, sem a necessidade de utilização de cartões de créditos.

Para utilizar essa modalidade, é necessário que a instituição de ensino possua um cadastro na AWS. Para maiores informações, acesse o seguinte link: <https://aws.amazon.com/education/awseducate/>

- **Cupons com créditos:** apesar de não ser necessariamente um tipo diferente de conta, é possível obter cupons com créditos para serem utilizados na AWS. Tais cupons são distribuídos em eventos promovidos pela Amazon. Com esses créditos, é possível utilizar os recursos da AWS, até que eles acabem.
- **Conta paga:** esse é o modelo de conta tradicional da AWS, onde empresas e usuários pagam pelos recursos que utilizam na AWS mensalmente.

## b) Criação da conta na AWS

Para aproveitar totalmente o conteúdo desse livro é necessário que o leitor crie uma conta na AWS, utilizando uma das opções citadas na seção anterior.

Tendo essa conta criada, independente da modalidade, será possível acompanhar todos os capítulos seguintes, criando e utilizando os recursos da AWS.

Para prosseguir com a criação de sua conta, acesse o seguinte link: <http://console.aws.amazon.com/>

Nessa tela, clique no botão *Create a new AWS account* e siga os passos que se apresentarem.

Depois que a conta for criada, será possível acessar o console de administração da AWS. Perceba que ele é uma interface Web rica em detalhes e opções ao usuário. À medida em que os conceitos forem apresentados ao longo dos capítulos, será possível aprender um pouco mais sobre as suas opções.

## c) Usuário IAM

A conta que foi criada na seção anterior funciona com um usuário administrador, dentro do console da AWS. Com ela já é possível criar e utilizar os recursos da AWS, porém, é recomendada a criação de usuários sob o domínio dessa conta. Dessa forma, é possível ter um controle maior, através de grupos e permissões, do que cada usuário pode criar ou acessar dentre os recursos da AWS.

Tais usuários podem ser criados utilizando um recurso gratuito da AWS chamado *Identity and Access Management*. O capítulo 4 trará maiores detalhes sobre esses usuários, principalmente como criá-los e atribuir permissões a eles.



Mesmo que a conta criada seja específica para acompanhamento desse livro, é altamente recomendado que o leitor opte pela criação de um usuário IAM para utilização dos recursos da AWS, conforme será detalhado no capítulo 4.

## d) AWS CLI

É possível criar e gerenciar os recursos da AWS através de uma aplicação, que pode ser instalada no computador, chamada *AWS Command Line Interface*.

A princípio, tudo o que pode ser feito no console Web da AWS pode ser feito no CLI, porém através de comandos com uma sintaxe específica.

O capítulo 4 trará os passos necessários para a configuração do AWS CLI, após o usuário IAM ser criado, juntamente com sua chave de acesso. Esses passos são necessários para que o AWS CLI possa ser ativado no computador que será utilizado para o desenvolvimento das aplicações apresentadas nesse livro.

## 2.2 - Ambiente de desenvolvimento

Para que o leitor possa acompanhar os capítulos desse livro, em sua totalidade, é necessário preparar seu ambiente de desenvolvimento, instalando algumas ferramentas e principalmente a IDE a ser utilizada para criação das aplicações de exemplo.

É importante observar que os passos descritos a seguir estão em linhas gerais do que deve ser feito, pois cada sistema operacional possui suas peculiaridades. Por isso, é necessário seguir os passos definidos nos links que serão fornecidos, de acordo com o sistema operacional que será utilizado.

### a) Java Development Kit

Antes de mais nada é necessário instalar o Java Development Kit (JDK), um conjunto de aplicações e bibliotecas necessário para se desenvolver aplicações em Java.

Esse livro está baseado no JDK 8. Ele pode ser encontrado através do seguinte link:

<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Para baixá-lo, é necessário aceitar os termos da licença de uso, escolher o sistema operacional e sua versão. Note que alguns sistemas operacionais podem ter passos adicionais para a configuração do JDK 8.

### b) IntelliJ IDEA Community Edition

Para desenvolver as aplicações de exemplo que serão criadas nesse livro, é necessário instalar um ambiente integrado de desenvolvimento, ou como é conhecido pela sua sigla em Inglês - IDE.

A IDE que será utilizada é o IntelliJ IDEA Community Edition, do fabricante JetBrains. Essa é uma das mais modernas IDE para desenvolvimento em Java e outras linguagens de programação.

Ele pode ser encontrado no seguinte endereço: <https://www.jetbrains.com/idea/download>

Baixe e instale a versão compatível com seu sistema operacional, seguindo as instruções definidas no site da JetBrains.

Caso o leitor queira aprender mais como trabalhar com o IntelliJ IDEA, há um excelente conjunto de tutoriais em vídeo na seguinte página de documentação do site da JetBrains: <https://www.jetbrains.com/idea/documentation>



## c) Repositório de código

Esse é um passo opcional, caso o leitor queira armazenar o código fonte dos projetos de exemplo em um repositório de código.

Uma excelente opção para isso é utilizar o GitHub, que permite a criação de repositórios públicos e privados sem a necessidade de se pagar uma mensalidade.

Para criar uma conta no GitHub, acesse o link a seguir e siga os passos para o preenchimento do formulário:

<https://github.com/join>

Para detalhes de qual plano escolher, acesse o seguinte link:

<https://github.com/pricing>

O plano gratuito é suficiente para acompanhamento desse livro.

## d) Cliente MySQL

Como algumas aplicações de exemplo irão utilizar uma base de dados relacional MySQL, será interessante ter um cliente para acessá-lo, ver suas tabelas e seus dados.

Existem várias aplicações clientes para esse tipo de trabalho. Uma excelente opção é o MySQL Workbench em sua versão gratuita. Ele pode ser baixado nesse endereço:

<https://dev.mysql.com/downloads/workbench/>

Selecione a opção adequada ao seu sistema operacional e siga as instruções de instalação oferecidas na página do fabricante.

No capítulo em que uma base de dados for utilizada por uma aplicação, haverá instruções específicas de como configurar o MySQL Workbench para se conectar a tal base. Por isso, não se preocupe em fazer passos adicionais. No momento, só é necessário instalar o cliente do MySQL Workbench.

## 2.3 ) Docker

Executar aplicações em *container* é uma tendência em *cloud computing* e também em aplicações distribuídas em micro-serviços. Esses são detalhes que ainda serão melhor explicados em capítulos a frente.

Como o Docker funciona e o no que ele ajudará ficará para alguns capítulos a frente, pois o intuito desse é apenas apresentar o que deve ser instalado.

## a) Conta no Docker

Um dos processos que será realizado ao longo desse livro é a criação de uma imagem de um *container* Docker para a execução da aplicação em um AWS ECS. A princípio parece algo complexo, mas na verdade é bem simples se alguns passos forem seguidos corretamente.

Para a utilização dessa imagem, é interessante colocá-la em um repositório para que a AWS possa baixa-lo e utiliza-lo. No momento, apenas crie uma conta no *hub* do Docker, no link a seguir:

<https://hub.docker.com>

Essa conta é gratuita e servirá totalmente para os objetivos desse livro.

## b) Docker

Da mesma forma como as demais aplicações apresentadas nesse capítulo, você deve instalar o Docker de acordo com a versão do seu sistema operacional. Para baixá-lo, veja as instruções contidas no link à seguir:

<https://docs.docker.com/install/>

Se quiser ir diretamente aos links para Mac e Windows, aqui estão:

- Mac: <https://docs.docker.com/docker-for-mac/install/>
- Windows: <https://docs.docker.com/docker-for-windows/install/>

É importante ler as instruções de preparação e instalação. Normalmente não há problemas em computadores mais novos, mas como o Docker trabalha com virtualização de hardware, assim como o VMWare e o VirtualBox, podem existir certas incompatibilidades ou configurações adicionais que deverão ser feitas.

## c) Repositório no Docker Hub

Depois que a conta foi criada no Docker Hub, será necessário criar um repositório para cada imagem que for sendo criada. Esse passo será detalhado mais adiante, quando a primeira imagem for criada.

Nesse momento, basta que o leitor tenha apenas a conta no Docker Hub.

## 2.4 - Postman

Postman é um aplicativo gratuito muito útil e versátil. Com ele será possível fazer requisições às aplicações desenvolvidas nesse livro, tanto quando estiverem em execução na própria máquina de desenvolvimento, bem como quando estiverem hospedadas na AWS.

Para baixá-lo, siga as instruções contidas nesse link: <https://www.getpostman.com>

Com o Postman é possível organizar e salvar requisições montadas nele em forma de coleções, organizadas por projeto ou por contexto.

Também é possível criar uma conta no Postman, para armazenar e compartilhar as coleções de requisições que forem sendo criadas nele. Isso facilita bastante o trabalho em uma equipe de desenvolvedores e testadores.

## 2.5 - Conclusão

Nesse capítulo foram introduzidas as ferramentas e pacotes a serem instalados, bem como as contas a serem criadas para que o leitor possa acompanhar esse livro em sua totalidade.

Obviamente, algumas configurações ainda deveram ser feitas, seja em aplicações ou dentro da conta da AWS e do Docker Hub. Porém, é mais interessante do ponto de vista didático que o leitor faça isso no momento em que o recurso tiver que ser utilizado.

O próximo capítulo detalha o que será desenvolvido nesse livro, em termos de projetos e aplicações, explicando o que será utilizado de recursos da AWS. Ele é importante para que o leitor possa ter uma noção clara do que vem pela frente, capítulo a capítulo.

## 3 - Conteúdo do livro

Neste capítulo você terá uma visão geral do que será desenvolvido ao longo do livro, utilizando os recursos que a AWS oferece, além das tecnologias que serão utilizadas para tais aplicações

Para cobrir todo o conteúdo planejado nesse livro, mais de um sistema ou contexto de aplicações será desenvolvido, com propósitos didáticos para poder cobrir os serviços da AWS que serão explicados nesse livro.

O **capítulo 4** dá uma visão geral sobre alguns conceitos como **contas**, **regiões**, **usuários** e **papéis**.

O **capítulo 5** dá uma introdução à construção de aplicações em **Java** utilizando o **Spring Boot**. Ao longo do livro serão desenvolvidas duas aplicações utilizando essa base, por isso é importante apresentar tais conceitos de forma a deixar o leitor com um nível mínimo de preparação.

O **capítulo 6** irá introduzir alguns conceitos do **Elastic Compute Cloud**, para criação de instâncias de máquinas para executarem aplicações. No exemplo a ser demonstrado, uma aplicação simples em Java será executada em duas instâncias do EC2, tendo um **application load balancer** controlando as requisições externas à aplicação, como pode ser visto na figura a seguir:

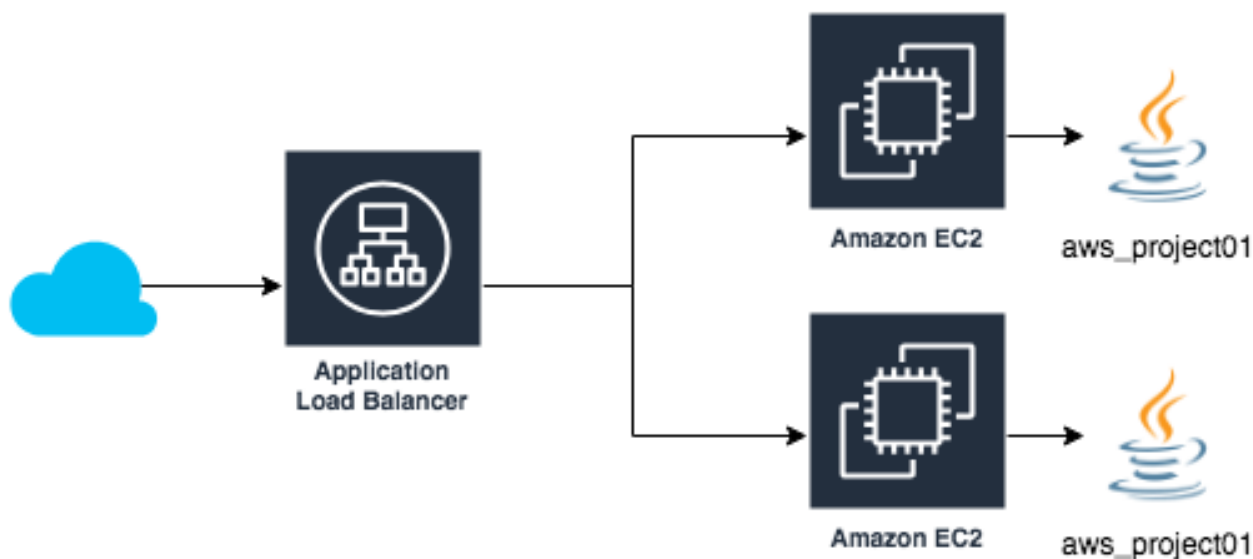


Diagrama do capítulo 6

No **capítulo 7** será apresentado um recurso muito interessante da AWS, o **CloudFormation**, que é capaz de criar recursos a partir de arquivos em formato YAML ou JSON. Nesse capítulo, instâncias EC2 serão criadas e configuradas a partir de um arquivo no formato JSON, que será processado e “executado” pelo AWS CloudFormation:

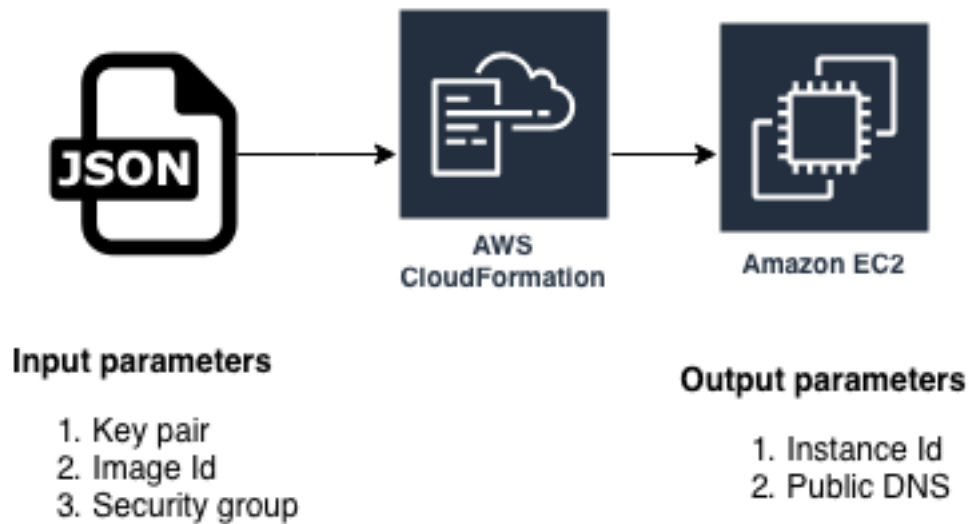
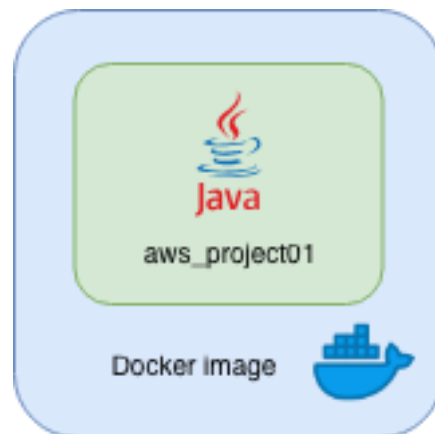


Diagrama do capítulo 7

O **capítulo 8** irá mostrar como colocar uma aplicação desenvolvida em Java e com Spring Boot para ser utilizada em *containers Docker*:



Aplicação aws\_project01 rodando em uma imagem Docker

Isso será essencial para a apresentação do **capítulo 9**, onde será introduzido o conceito do **Amazon Elastic Container Service**, responsável pela criação de *clusters* para execução de tarefas de aplicações baseadas em *containers Docker*:

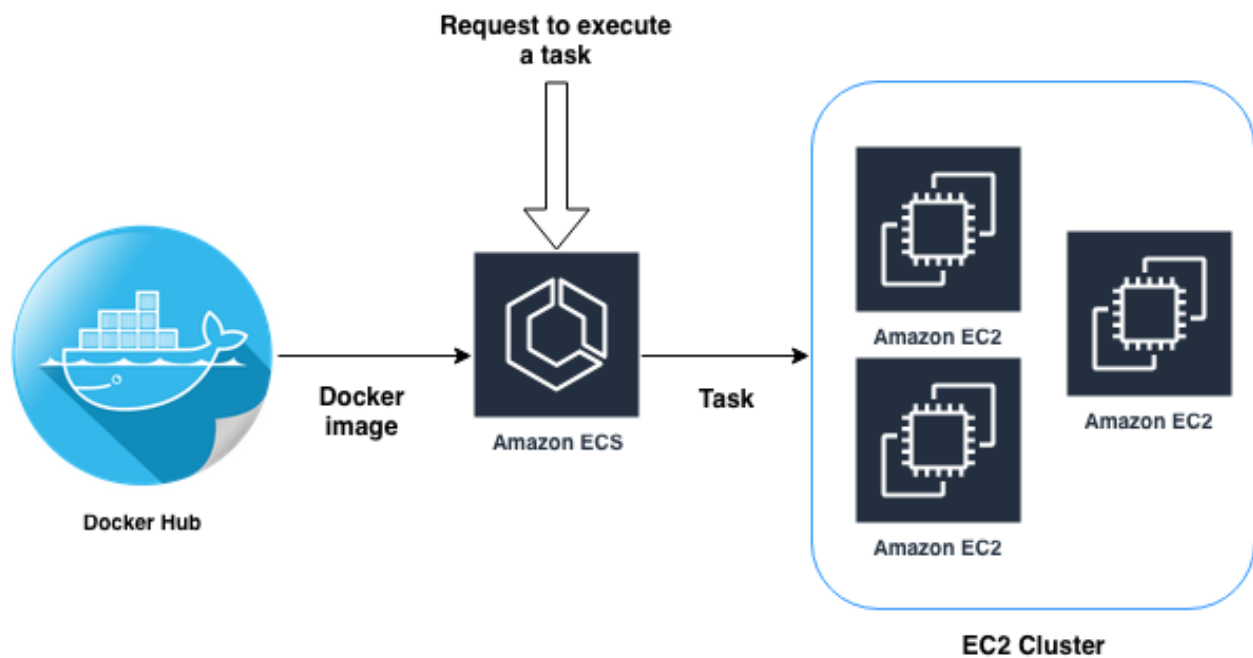
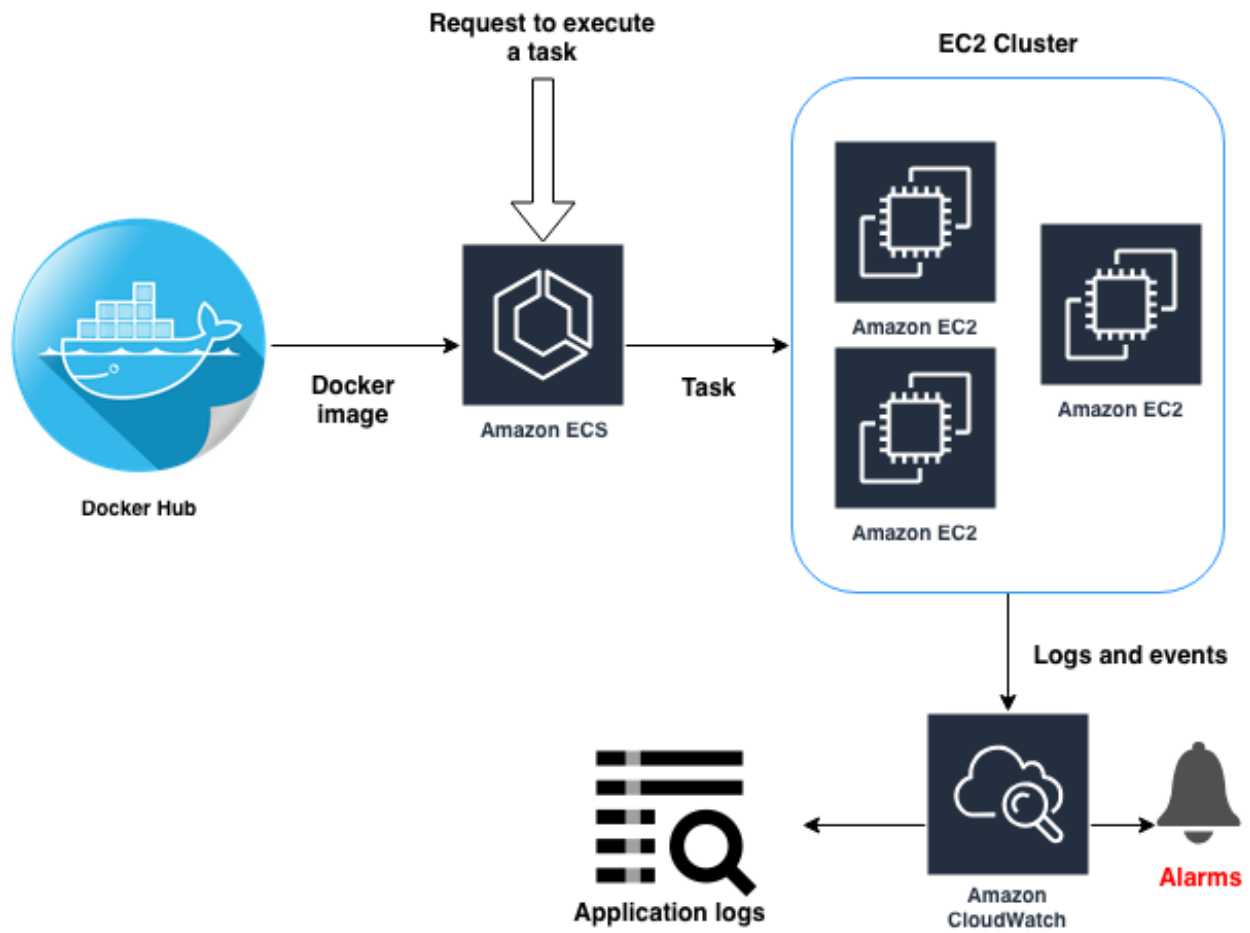


Diagrama do capítulo 9

Nesse capítulo serão apresentados os conceitos de definições de tarefas e serviços a serem executados no ECS, através de *clusters* de instâncias EC2, capazes de executar mais de um serviço ao mesmo tempo, compartilhando os recursos oferecidos pelo *cluster*.

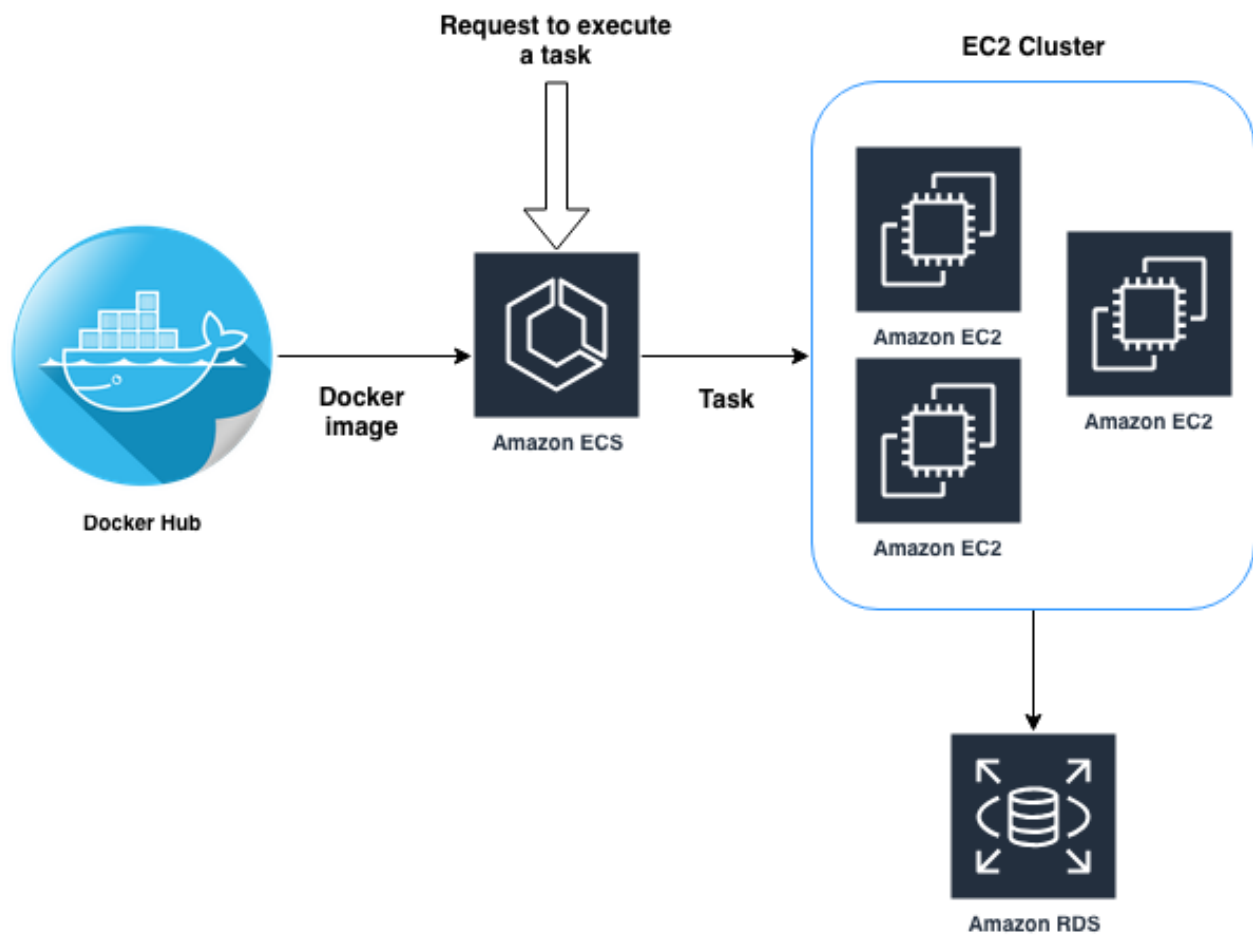
O **capítulo 10** apresentará um poderoso serviço de monitoramento da AWS, o **CloudWatch**, que permite a concentração de logs de aplicações, execução de outros recursos AWS, além da configuração de eventos e visualização de métricas de serviços através de gráficos.





CloudWatch e a aplicação aws\_project01

O capítulo 11 mostrará como criar, utilizar e gerenciar instâncias de banco de dados no AWS Relational Database Service.



Acessando um banco de dados com o RDS

Com ele, será possível fazer com aplicações possam persistir seus dados em uma instância de um banco de dados, como o MySQL.

O **capítulo 12** explica como utilizar o **AWS Simple Notification Service**, um serviço que permite que aplicações publiquem notificações em tópicos.

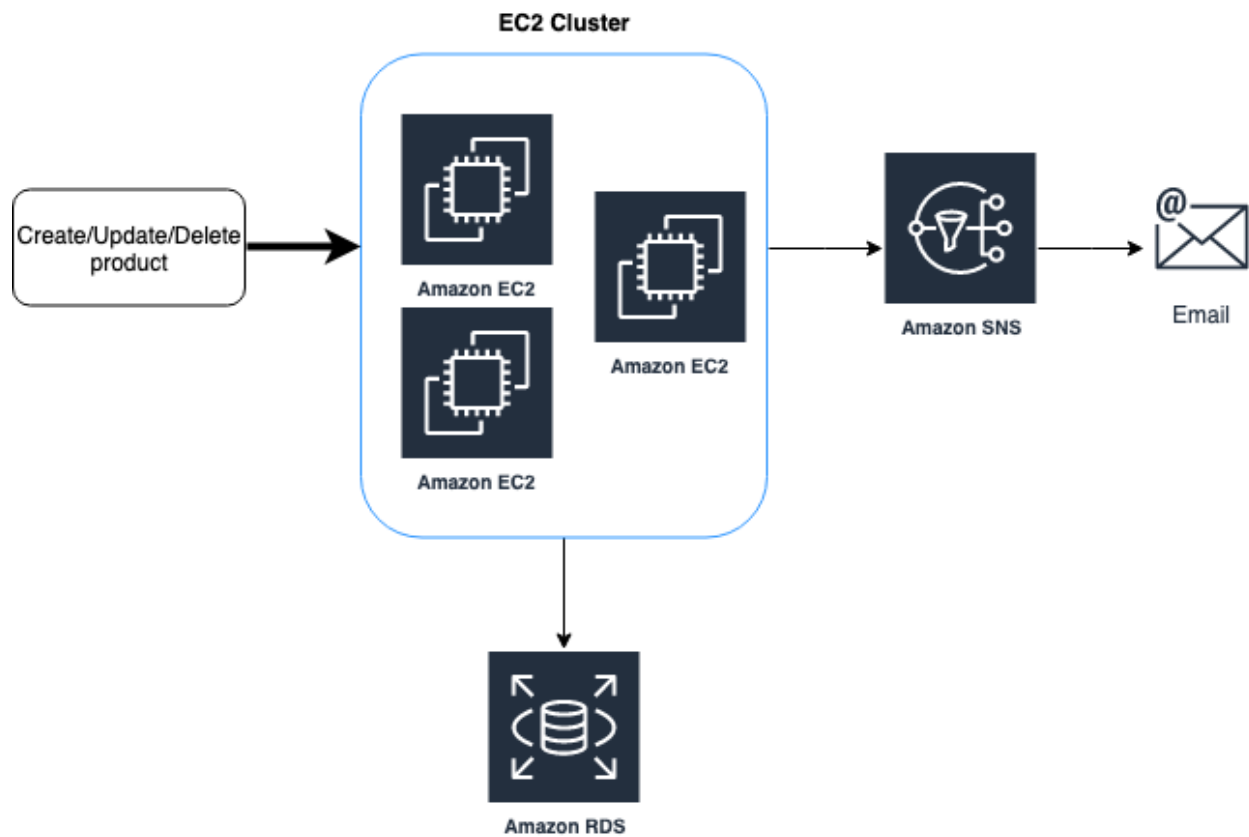
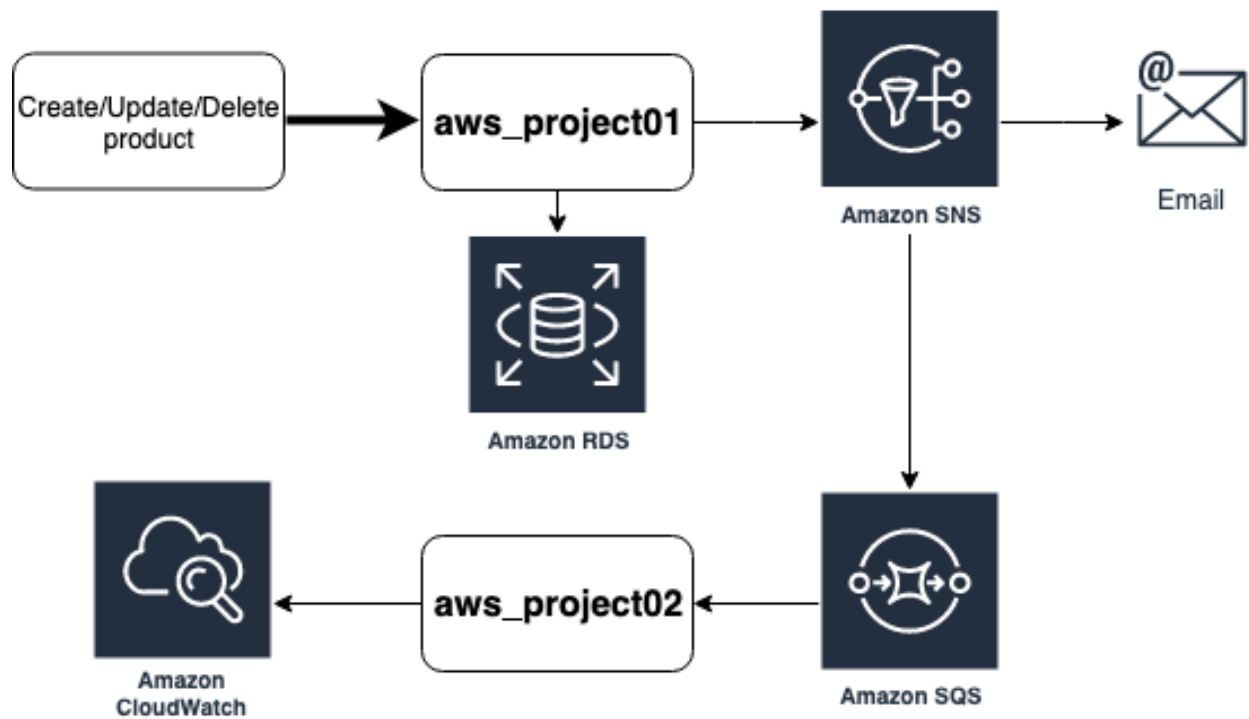


Diagrama da aplicação com SNS

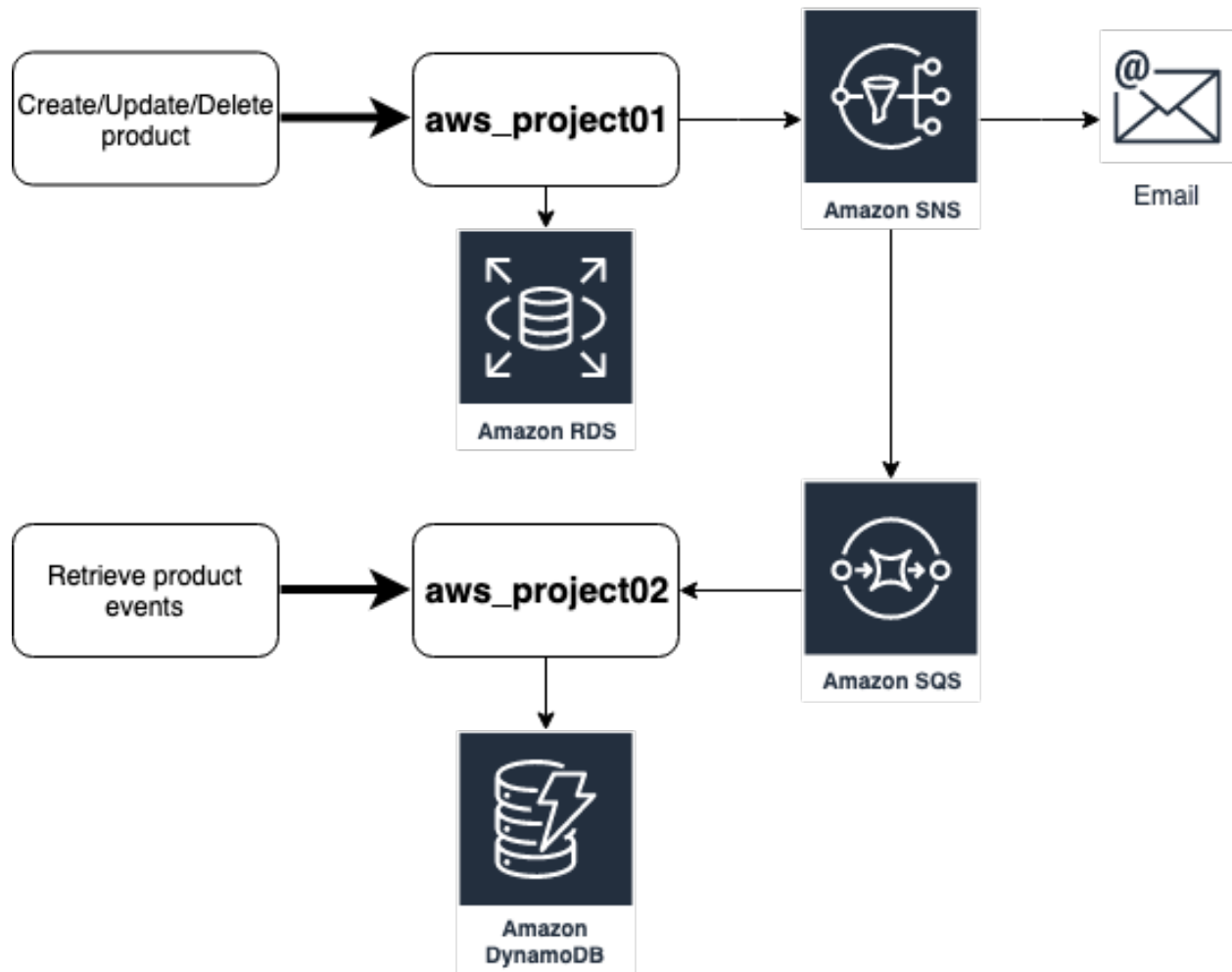
Esses tópicos podem entregar mensagens para endereços de e-mail ou podem ser utilizados para publicar mensagens em filas, como o **AWS Simple Queue Service**, como será visto no **capítulo 13**:



Integração com SNS e SQS

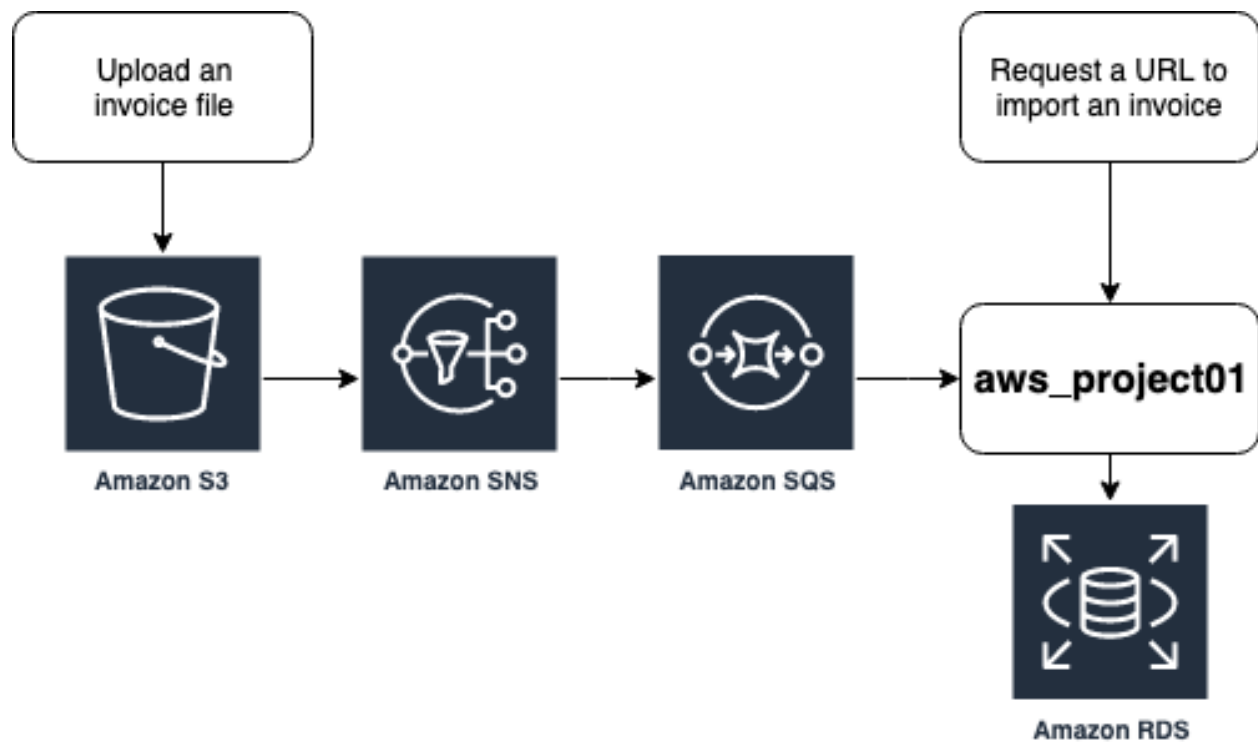
Dessa forma, é possível criar uma comunicação de forma **assíncrona** entre duas aplicações.

No **capítulo 14**, será detalhado conceitos iniciais do AWS **DynamoDB**, um serviço de banco de dados não relacionais, que permite a criação de itens que pode ser armazenados em forma de chave-valor e também em documentos.



Salvando eventos no DynamoDB

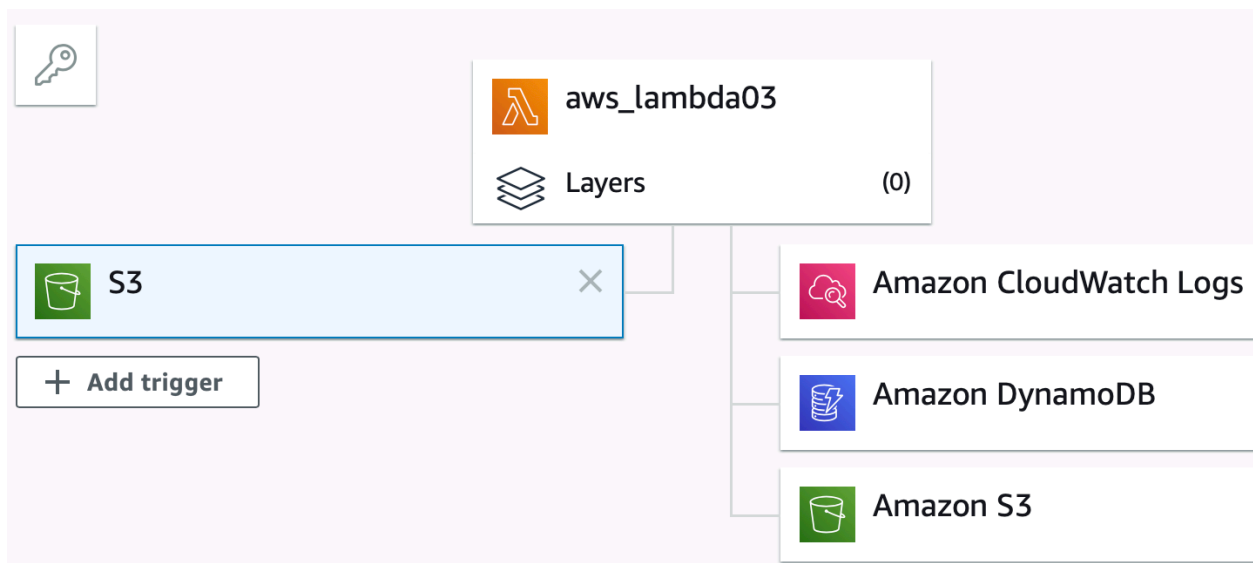
O capítulo 15 irá mostrar como utilizar o **AWS Simple Storage Service**, para armazenamento de arquivos, que podem ser gerenciados através de aplicações sendo executadas em **clusters**:



#### Importando arquivos com o S3

O exemplo que será desenvolvido conta com uma integração entre os serviços S3, SNS, SQS, ECS e RDS, mostrando como uma aplicação pode ser notificada sobre a inserção de um novo arquivo em um **bucket** do S3.

Finalmente, no capítulo 16, será apresentado conceitos de como construir aplicações *serverless* com o **AWS Lambda**.



Função Lambda acessando S3 e DynamoDB

Alguns exemplos serão desenvolvidos, mostrando como funções Lambdas podem ser utilizadas para as mais variadas aplicações. Na figura anterior mostra um diagrama de uma função Lambda acessando um *bucket* S3 e salvando os dados do arquivo em uma tabela do DynamoDB, além de gerar logs de sua execução no CloudWatch Logs.

Aproveite essa jornada de conhecimento e, caso necessite de ajuda, não hesite em chamar o autor: [siecola@gmail.com](mailto:siecola@gmail.com)

Divirta-se!