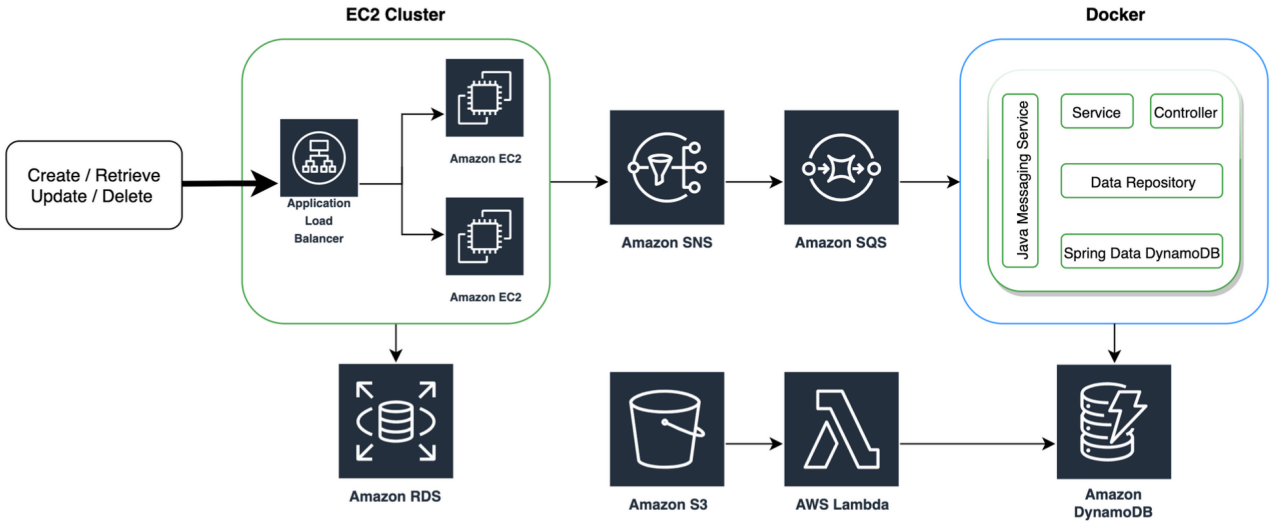
First edition

siecola.com.br

Developing Java applications on AWS



PAULO SIECOLA

Developing Java applications on AWS

Create and deploy Java apps with Spring Boot and Docker on AWS ECS - Elastic Container Service

Paulo Cesar Siecola

This book is for sale at http://leanpub.com/amazonwebservice_en

This version was published on 2021-04-03

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

For Joana and Rogério, my devoted parents. Special thanks to Isabel Mendes, my wife, for the excellent technical review.

Contents

1 - Introduction

Welcome to the **Developing Java Applications on AWS** book! It will teach you how to develop applications that will run in the Amazon Web Services environment. Also, the reader will understand how various cloud services work and their interactions through applications that will be built throughout the chapters.

The reader who has decided to follow this book might already know the relevance that cloud computing application development has in various areas of technology and business. Much more than a trend, this approach to building systems and software applications has become necessary for many business areas. Currently, few segments still use their own infrastructure to host their applications, perhaps still motivated by insecurity and lack of knowledge in cloud computing infrastructure.

To follow this book, the reader must have prior knowledge of application development using the **Java language and object-oriented programming**. Language-specific knowledge such as frameworks and development tools will be presented throughout the book.

Other languages that appear will be used minimally for building scripts or configuration files, necessary for building applications and configuring the services that will be created.

The applications taught in this book will use modern frameworks and tools, such as **Spring Boot** and **Docker**. In this way, the reader will learn to work with such technologies in conjunction with cloud computing services.

1.1 - Who is this book for?

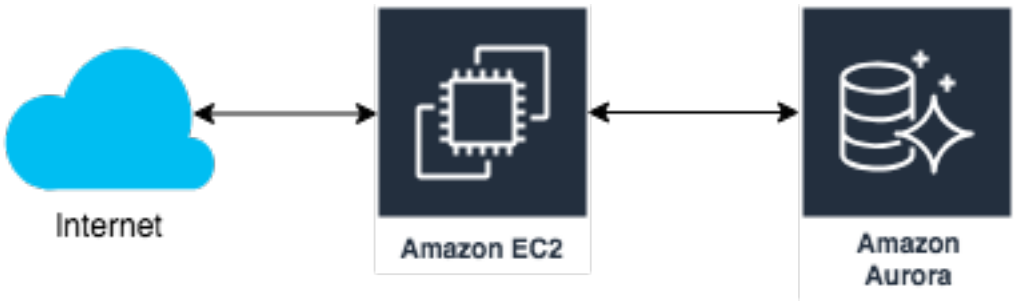
This book is intended for **software developers**, with or without experience in cloud computing and web development, who want to dig deeper into the services offered by Amazon Web Services.

It can also serve as a source of knowledge for **system administrators and operators** to familiarize themselves with Amazon Web Services cloud resource creation and administration tools.

1.2 - What is AWS

AWS stands for **A**mazon **W**eb **S**ervice, Amazon's cloud computing platform. Here one can build multi-purpose applications hosted on a robust and scalable infrastructure using services such as databases, servers, data storage devices, and more.

A typical example of what one can do using AWS is to create a virtual machine to run an application and connect it to a database to persist information, as shown in the following figure:



Database Application Example

The two features shown in the figure are AWS services and will be detailed below, but they represent an instance of a virtual machine (Amazon EC2) where the application runs and the database service (Aurora) to persist application information.

At this point, there is nothing new, but these features (the virtual machine and the database server) can be turned on or off as needed and can be modified in terms of processing and storage as the application demand grows or shrinks. Thus, the cost of using such resources can be adjusted according to the number of requests the application receives.

This feature of on-demand utilization of cloud computing resources is one of the significant advantages of this technology. Costs of using resources and services are somewhat accompanying variations in their usage, making their use attractive from a business standpoint and more financially viable.

Resources can be accessed directly from the Internet or protected with **security rules** based on various configuration parameters, such as being accessible only through other AWS resources, i.e., completely blocked from the "outside world".

Security of access to resources and information is fundamental in cloud applications, and a feature strongly present in the services offered by AWS.

1.3 - AWS services and resources

This session will introduce some of the **services and features offered by AWS**, **especially those covered in this book**.

The intention is to briefly present some services with their nomenclatures and symbols, which will be used in the diagrams presented throughout the book, so the reader gets used to them when they appear.

All of the features presented here will be covered later in the book throughout its chapters. However, it is possible that, for didactic reasons, some resources are cited before they are formally presented, so a brief introduction to them is important.

a) Amazon Elastic Compute Cloud - EC2

Maybe this is the most popular service that AWS provides. With it, one can create **virtual machine instances** with different processing capacity, memory, and storage.

During the process of creating an EC2 instance, one can choose the operating system to run on the virtual machine. With this, after its creation, the virtual machine will be ready for use.

The symbol used to represent an EC2 instance in diagrams is the following:

AWS EC2

With an EC2 instance, one can, for example, install a server and host a web application available across the Internet. One can also run an application inside a container using technologies like Docker, which will be used and explained throughout the chapters of this book.

b) Amazon Elastic Container Service - ECS

With ECS one can run applications inside **containers such as Docker**. Thus, one does not need to take care of administering instances of EC2 virtual machines that would host such applications.

Within an ECS, the system administrator can create tasks to run multiple virtual machines, causing the same application to have multiple instances running at the same time. This way, it's possible to balance the processing and the requests between the various instances that are running the application. Also, it is possible to increase or decrease the number of instances automatically, according to such processing parameters and the number of requests.

The symbol representing an ECS is the following:

AWS ECS

Running applications within containers have many advantages, especially concerning orchestrating the number of application instances to be executed and where traffic should be directed, in case of large requests or processing. This is a subject that will be addressed in this book as the concept of ECS is explored in more detail.

c) Amazon CloudWatch

CloudWatch is an Amazon Web Service monitoring and management service. Some of its most common functions are:

- Visualization of logs generated by other resources and applications;
- **Metric** graphs such as memory consumption and processing;
- **Events** visualization.

The symbol that represents CloudWatch is the following:

Amazon CloudWatch

In a system with applications distributed across multiple instances or with different types of resources, it is crucial to have a way of concentrating this information in one place for easier monitoring.

d) Amazon Relational Database Service - RDS

This is a service for creating **relational databases** on AWS, such as **Aurora**, **MySQL**, **PostgreSQL**, **Oracle**, and **SQL Server**, without the need for server administration and operation.

The symbol representing RDS is the following:

Amazon RDS

RDS also offers benefits such as:

- Backup scheduling;
- Data recovering;
- Data replication;
- Monitoring and metrics.

e) Amazon DynamoDB

Amazon DynamoDB is a database based on **documents** and **key-value structures**. Quite unlike a relational database, in terms of data storage structure, this service has high-speed access operations at any scale.

The symbol representing the Amazon DynamoDB is the following:

AWS DynamoDB

Some of its features are:

- Backup scheduling;
- Data recovering;
- A mechanism that allows notifying the application in case of records changing;
- High performance for writing and reading operations;
- A mechanism for automatic records deletion.

f) Amazon Simple Queue Service - SQS

SQS is the AWS's **message queuing mechanism**, for example, to establish communication between different applications.

The symbol that represents Amazon SQS is the following:

AWS SQS

Following are some of the features of SQS:

- Guaranteed delivery of at least one message;
- Has mechanisms to prevent message order inversion.

g) Amazon Simple Notification Service - SNS

With AWS SNS one can send **notification** messages to mobile devices such as Android and iOS, as well to trigger events for other AWS features. It can also work seamlessly with AWS SQS to allow message propagation from one application to several at the same time.

The symbol representing the Amazon SNS is the following:

AWS SNS

Here are some of its features:

- Allows integration with other AWS features such as SQS, Lambda, and S3;
- Several SQS can register in an SNS to receive notifications;
- Message protocol independency.

h) Amazon Simple Storage Service - S3

The AWS S3 is one of its **mass object storage** services. It is mainly used for temporary or non-temporary file storage with high access availability.

The symbol representing AWS S3 is the following:

AWS S3

Here are some of its features:

- High availability;
- It's possible to send a notification to other AWS features when a new object is inserted;
- Has security settings to restrict public or private access.

i) AWS Lambda

Lambda is the AWS's **serverless code execution service**.

With it, one can react to notifications or events and run a small chunk of code quickly, without having to maintain an idle virtual machine instance or server infrastructure.

The symbol that represents it is the following:

AWS Lambda

Some of the key features of AWS Lambda are:

- There is no need to create servers to execute source code;
- Multiple instances can be run to meet any request demand;

- The cost of using it is charged per code execution time, i.e., only when something needs to be executed.

AWS offers some many other features and services. For an up-to-date list, see the following link: https://aws.amazon.com

1.4 - Didactic structure of the book

The didactic structure of this book permeates a concept known as **problem-based learning**, where the reader is introduced and led to key concepts through problems that must be solved using such knowledge.

Of course, not all concepts can be presented this way, but for those who allow, a more practical approach to learning will be used, so that the reader always has in mind a problem to be solved using the technology that is presented in detail. Thus, keeping in mind the target problem to be solved, the reader can absorb the concepts that are most efficiently presented.

1.5 - What will be needed

The next chapter details everything the reader will need to follow this book. Here is a summary so one can see that nothing unusual or costly will be used:

- Computer with Windows, OS X or Linux;
- Free development tools;
- AWS user account, which can be created in various ways for a free trial period;
- Code repository on GitHub.

The Postman tool will be widely used in this book to access REST services that will be created. The reader must be familiar with such a tool.

1.6 - AWS feature limitations by account type

There are certain resource restrictions that AWS imposes, depending on the type of account one use to access it. An example of this is the AWS Educate[1] program, which offers credits for students and teachers, but which limits some resources or features.

These restrictions are under constant reviewing by AWS, so please check the ones that are imposed on your chosen account type.

[1] https://aws.amazon.com/education/awseducate/

1.7 - Next steps

In this chapter, the reader had a brief introduction to what AWS is, Amazon's cloud computing service, and some of the features it offers. Also, it was presented as the book is structured.

In the next chapter, the reader will look at what to do to prepare his development environment and also the first steps that must be taken in the AWS administration console so that he can start building resources and developing applications.

2 - Preparing the environment

This chapter describes everything that should be installed or prepared to follow this book:

- AWS Account;
- Development Environment with IntelliJ IDEA
- Other tools, such as Postman, which is a REST client.

2.1 - AWS console

Amazon has an administration console to use AWS, create, manage resources, and then run applications.

This console can be accessed at the following address: http://console.aws.amazon.com/

a) Amazon Web Services account types

To access AWS through its console, the reader must create an account first. This way, it will be possible to control the use of resources, expenses (when applicable), monitor utilization alarms, among many other features.

AWS has some types of accounts described below. Any of these accounts can be used to follow this book:

- **Free tier**: maybe this is the simplest way to start using AWS for learning purposes. With such an account the reader can have 12 months free to use some of the AWS features within specified limits.

  AWS constantly renews the features and limits of this type of account. For an up-to-date view of this modality, please visit the following link: https://aws.amazon.com/free

  This requires the use of a credit card for the account creation, but it won't charge it if the resource usage is under the limits set for the AWS free tier criteria.
- **AWS Educate**: this is an ideal choice for faculty and students in schools, colleges, and universities, as it offers multiple monthly credit schemes to users without the need to use credit cards.

  To use this modality, the educational institution must have an AWS registration. For more information, please visit the following link: https://aws.amazon.com/education/awseducate/
- **Coupons with credits**: while this is not necessarily a different type of account, the reader can get credit coupons for use on AWS. Such coupons are distributed at events promoted by Amazon. With these credits, the reader can use AWS resources until they run out.
- **Paid account**: this is the traditional AWS account model, where businesses and users pay for the features they use on AWS monthly.

b) AWS account creation

To fully leverage the content of this book, the reader must create an AWS account using one of the options mentioned in the previous section.

Having this account created, regardless of modality, the reader will be able to follow all the following chapters, creating and using AWS resources.

To proceed with the creation of the account, please visit the following link: http://console.aws.amazon.com/

On this screen, click the Create a new AWS account button and follow the steps that appear.

Once the account is created, the reader will be able to access the AWS Admin Console. Note that it is a web interface rich in detail and user options. As concepts are presented throughout the chapters, the reader will be able to learn a little more about these options.

c) IAM user

The account that was created in the previous section works as an administrator user within the AWS console. With it, one can create and use AWS features, but it is recommended to create users under the domain of this account. This gives the account administrator greater control, through groups and permissions.

Such users can be created using a free AWS feature called Identity and Access Management. Chapter 4 will go into more detail about these users, especially how to create them and assign permissions to them.

Even if the account created was specific to follow this book, it is highly recommended that the reader choose to create an IAM user to use AWS features, as detailed in Chapter 4.

d) AWS CLI

One can create and manage AWS resources through a computer-installable application called AWS Command Line Interface.

In principle, anything that can be done on the AWS web console can be done on the CLI, but with commands with a specific syntax.

Chapter 4 will give the steps required to configure AWS CLI after the IAM user is created along with the passkey. These steps are needed to enable AWS CLI to be enabled on the computer that will be used to develop the applications presented in this book.

2.2 - Development environment

To follow this book, it is necessary to prepare its development environment, installing some tools and mainly the IDE to be used for the creation of the sample applications.

It is important to note that the following steps outline what should be done, but each operating system has its peculiarities. Therefore, it is necessary to follow the steps defined in the links that will be provided, according to the operating system that will be used.

a) Java Development Kit

First of all, it's necessary to install the Java Development Kit (JDK), a set of applications and libraries needed to develop Java applications.

This book is based on JDK 8. It can be found through the following link:

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

To download it, one must accept its license, choose the operating system and its version. Note that some operating systems may have additional steps to setting up JDK 8.

b) IntelliJ IDEA Community Edition

To develop the sample applications that will be created in this book, the reader needs to install an integrated development environment.

The IDE that will be used is the IntelliJ IDEA Community Edition, from the manufacturer JetBrains. This is one of the most modern IDE to develop applications in Java and other programming languages.

It can be found at the following address: https://www.jetbrains.com/idea/download

Download and install the version compatible with the development machine operating system by following the instructions set on the JetBrains website.

If the reader wants to learn more about working with IntelliJ IDEA, there is an excellent set of tutorials on the following documentation page of the JetBrains website: https://www.jetbrains.com/idea/documentation/

c) Source code repository

This is an optional step if the reader wants to store the source code of the sample projects in a repository.

An excellent option for this is to use GitHub, which allows the creation of public and private repositories without the need to pay a monthly fee.

To create a GitHub account, go to the following link and follow the steps to complete the form:

https://github.com/join

For details of which plan to choose, visit the following link:

https://github.com/pricing

The free plan is sufficient to follow up on this book.

d) MySQL client

Since some sample applications will use a MySQL relational database, it will be interesting to have a client to access it, to see its tables and its data.

There are several client applications for this type of work. An excellent option is MySQL Workbench in its free version. It can be downloaded from this address:

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

Select the option appropriate for development machine operating system and follow the installation instructions offered on the manufacturer's page.

In the chapter where an application uses a database, there will be specific instructions on how to configure MySQL Workbench to connect to that database. So do not worry about taking additional steps here. At this time, the reader only need to install the MySQL Workbench client.

2.3 - Docker

Running container applications is a trend in cloud computing as well as distributed applications in microservices. These are details that will be explained in later chapters.

How Docker works and what it will help, will be covered in a few chapters ahead. The purpose of this section is to present what should be installed or created.

a) Docker account

One of the processes that will be done throughout this book is creating an image of a Docker container for running the application on an AWS ECS. At first, it sounds complicated, but it is quite simple if a few steps are followed correctly.

To use this image, it is interesting to put it in a repository so that AWS can download and use it. At this time, create an account on the Docker hub at the following link:

https://hub.docker.com

This account is free and will adequately serve the purposes of this book.

b) Docker

As with the other applications presented in this chapter, the reader should install Docker according to the development machine operating system version. To download it, see the instructions in the following link:

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

If the reader wants to go directly to the links for Mac and Windows, here they are:

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

It is important to read the preparation and installation instructions. Usually, there are no problems with newer computers, but since Docker works with hardware virtualization, just like VMWare and VirtualBox, there may be certain incompatibilities or additional configurations that need to be made.

c) Docker Hub repositories

Once the account has been created in Docker Hub, the reader will need to create a repository for each created image. This step will be detailed later when the first image is created.

At this point, it is enough for the reader to have only the Docker Hub account.

2.4 - Postman

Postman is a very useful and versatile free application. With it, the reader will be able to make requests to the applications developed in this book, both when they are running on the development machine itself as well as when they are hosted on AWS.

To download it, follow the instructions at this link: https://www.getpostman.com

With Postman, the reader can organize and save requests assembled on it in the form of collections, organized by project or by context.

One can also create a Postman account to store and share the collections of requests created on it. This makes working on a team of developers and testers much more straightforward.

2.5 - Conclusion

This chapter introduced the tools and packages to be installed, as well as the accounts to be created so that the reader can follow this book in its entirety.

Of course, some configurations have yet to be made, either in applications or within the AWS and Docker Hub account. However, it is more interesting from a didactic point of view that the reader does this when the resource has to be used.

The next chapter details what will be developed in this book, in terms of projects and applications, explaining what will be used of AWS resources. It is important for the reader to have a clear sense of what lies ahead, chapter by chapter.

3 - Book content

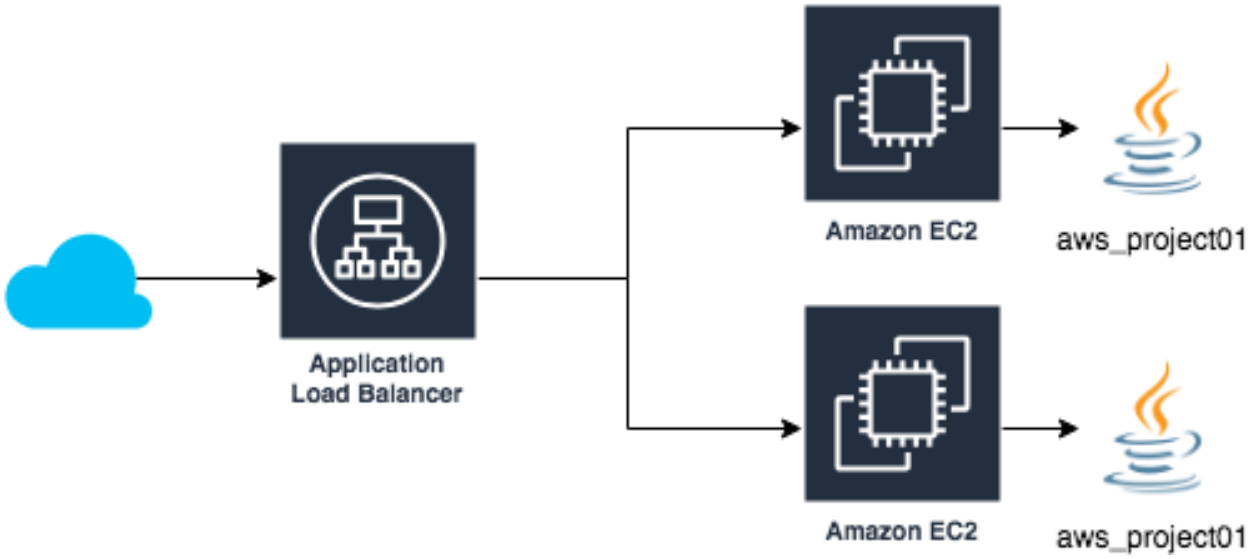
In this chapter, one will get an overview of what will be developed throughout the book, utilizing the features AWS offers, as well the technologies that will be used for such applications.

To cover all of the content planned in this book, more than one system or context of applications will be developed for teaching purposes to cover the AWS services that will be explained in this book.

Chapter 4 gives an overview of some concepts, such as **accounts**, **regions**, **users**, and **roles**.

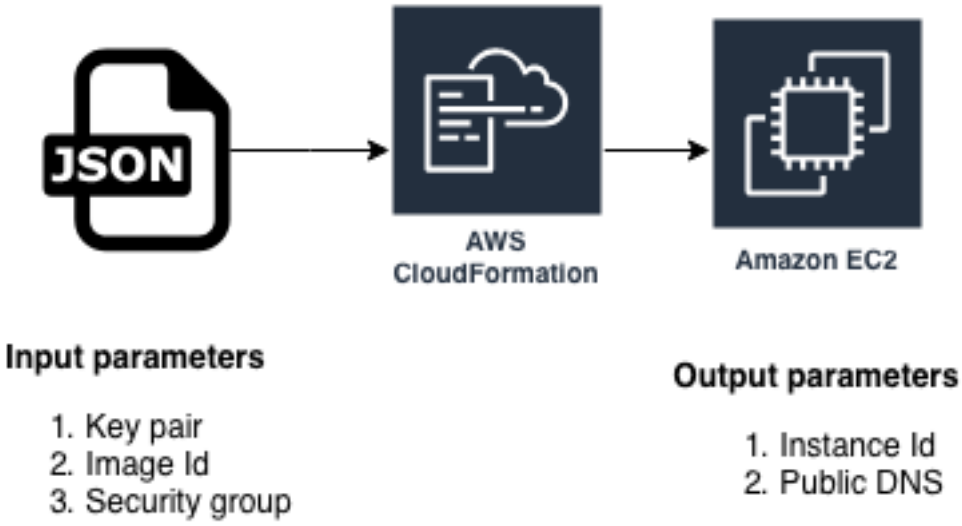
Chapter 5 gives an introduction to building **Java** applications using **Spring Boot**. Throughout the book, two applications will be developed using this foundation, so it is important to present such concepts to leave the reader with a minimum level of preparation.

Chapter 6 will introduce some **Elastic Compute Cloud** concepts for creating machine instances to run applications. In the example to be shown, a simple Java application will run on two EC2 instances, with an **application load balancer** controlling requests outside the application, as shown in the following figure:



Chapter 6 diagram

Chapter 7 will introduce an exciting feature of AWS, **CloudFormation**, which can create resources from YAML or JSON format files. In this chapter, EC2 instances will be created and configured from a JSON format file, which will be processed and executed by AWS CloudFormation:

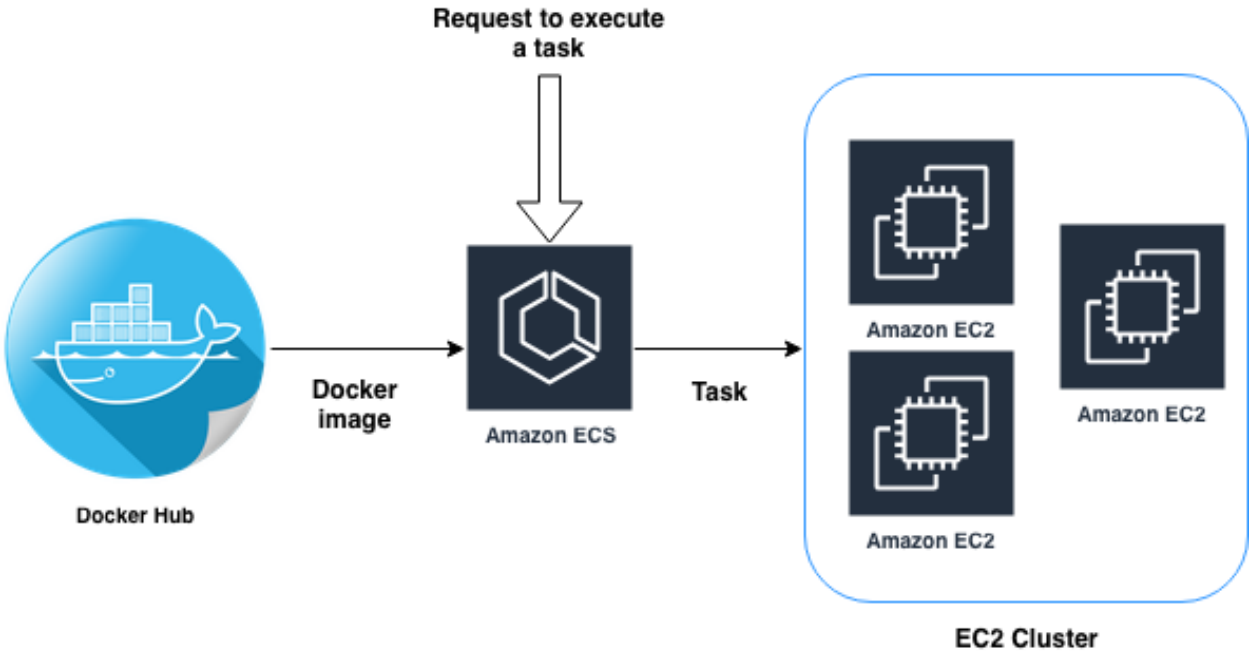


Chapter 7 diagram

Chapter 8 will show how to run a Java-based Spring Boot application in **Docker** containers:

aws_project01 application running in a Docker container

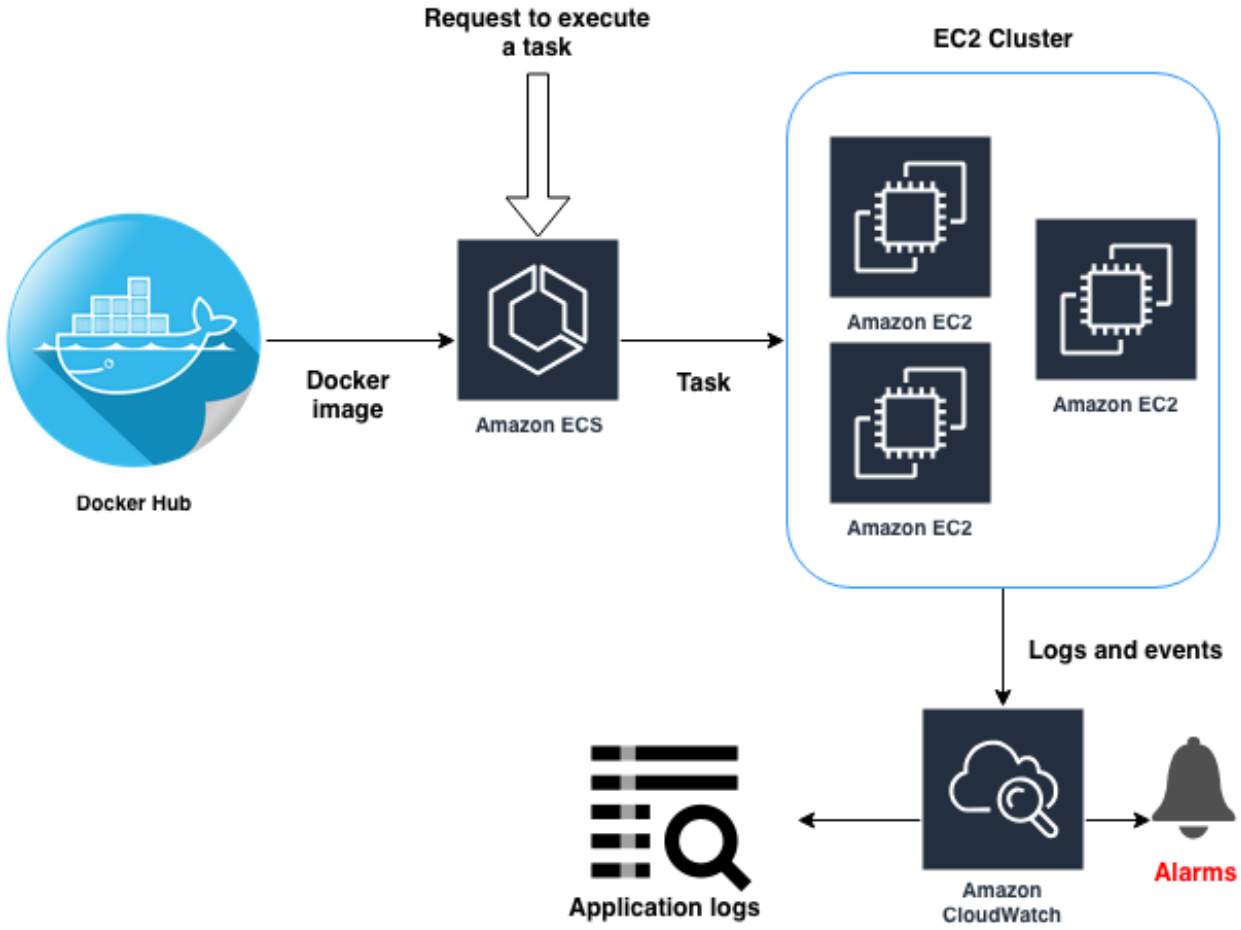
This will be essential for the introduction to **Chapter 9**, where the concept of the **Amazon Elastic Container** Service will be introduced, responsible to create clusters for Docker container-based application tasks:



Chapter 9 diagram

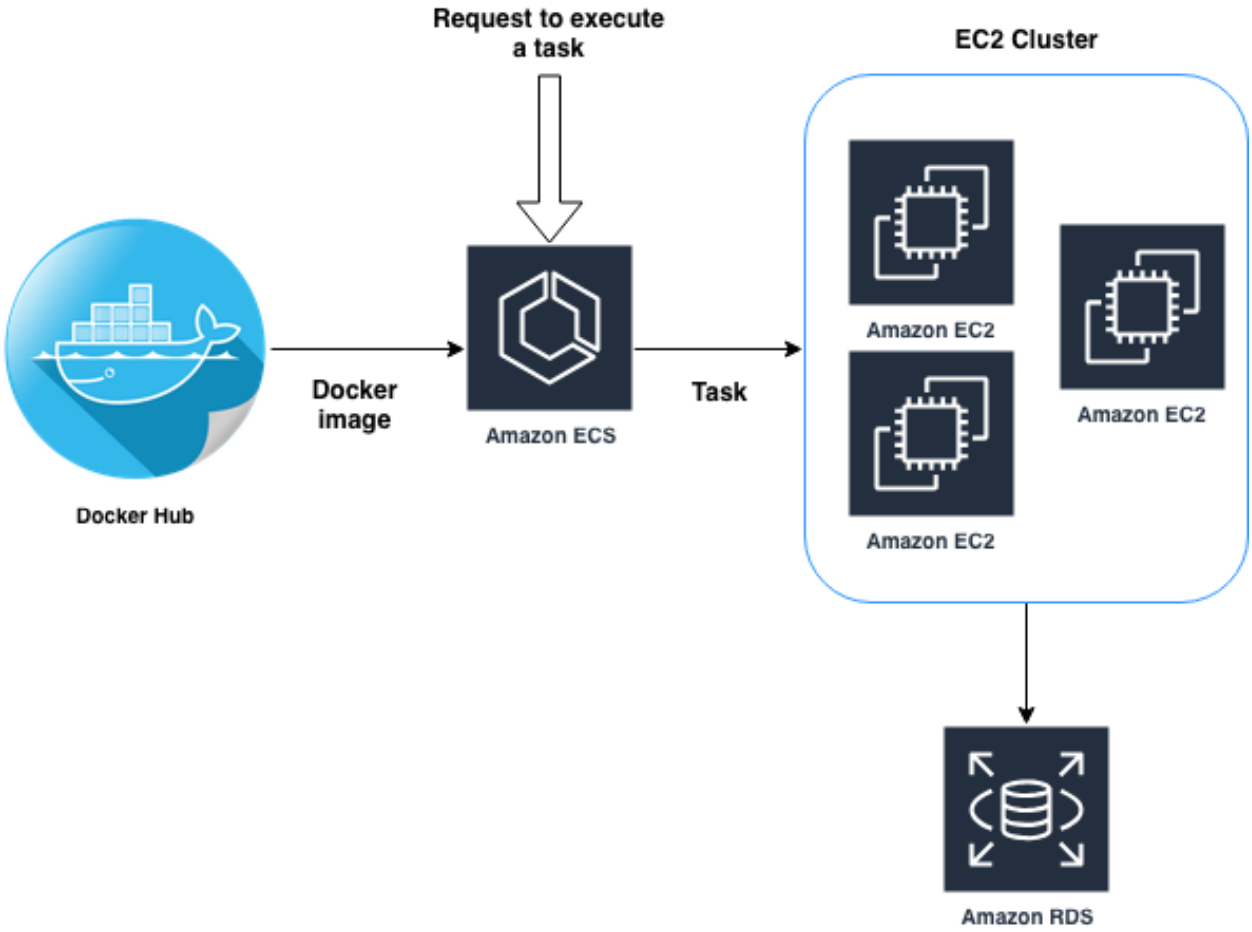
This chapter will present the concepts of task and service definitions used by ECS, through clusters of EC2 instances, capable of running more than one service at a time, sharing the resources offered by the cluster.

Chapter 10 will introduce a powerful AWS monitoring service, **CloudWatch**, which enables application logging, event configuration, and service metrics visualization through graphs.



CloudWatch and the aws_project01 application

Chapter 11 will show how to create, use, and manage database instances in AWS **Relational Database Service**.



Accessing a database in the RDS

This will make it possible for applications to persist their data in an instance of a database such as MySQL.

Chapter 12 explains how to use AWS **Simple Notification Service**, a service that lets applications publish notifications on topics.

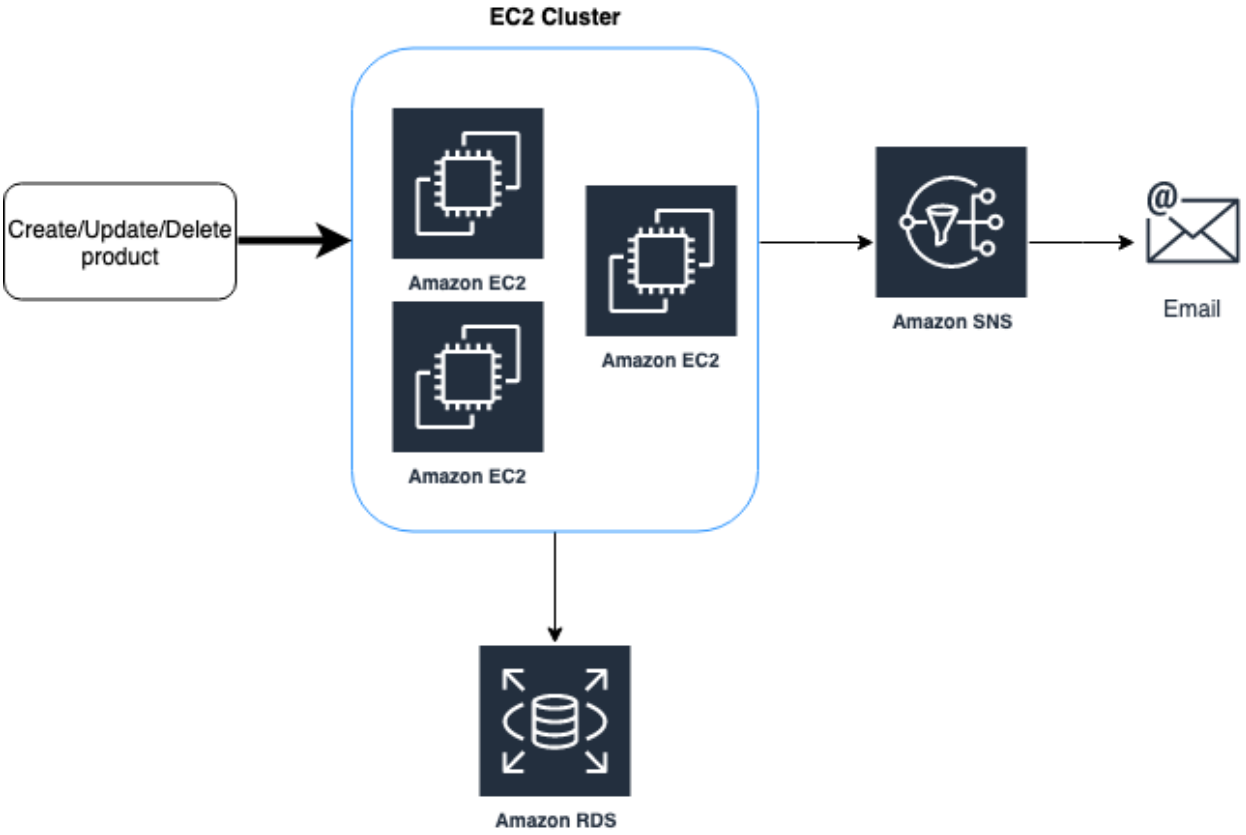
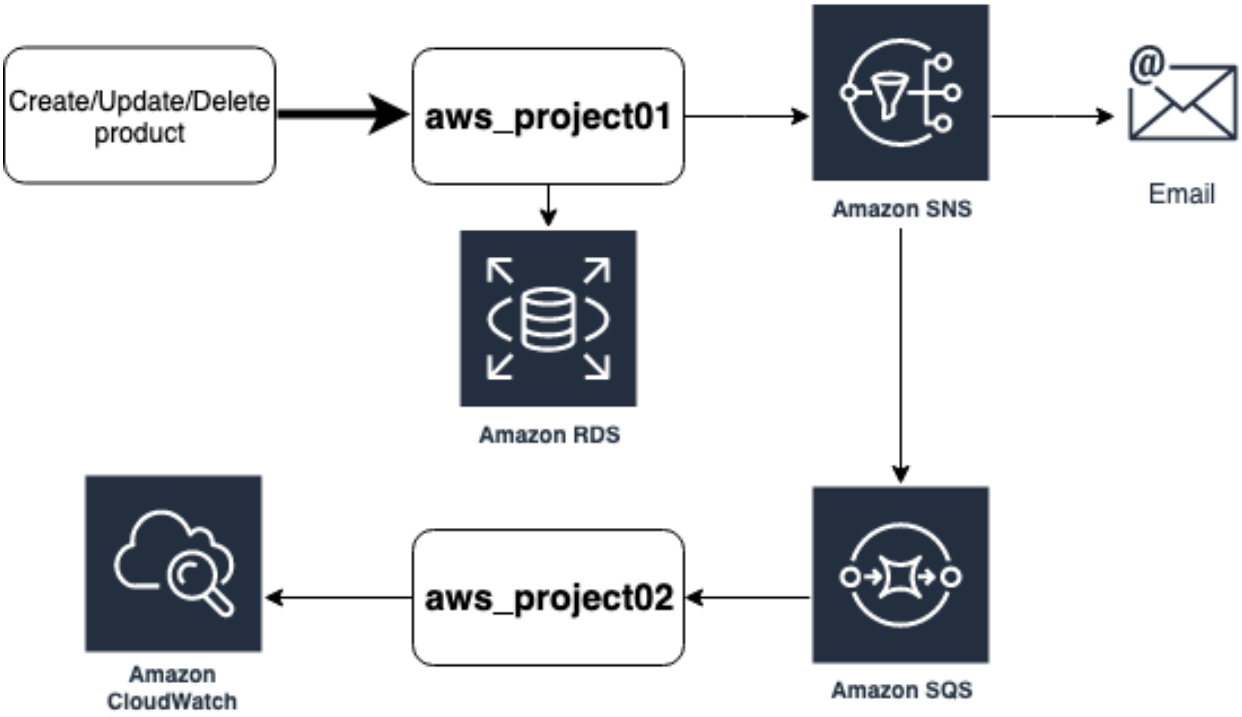


Diagram of the application with a SNS

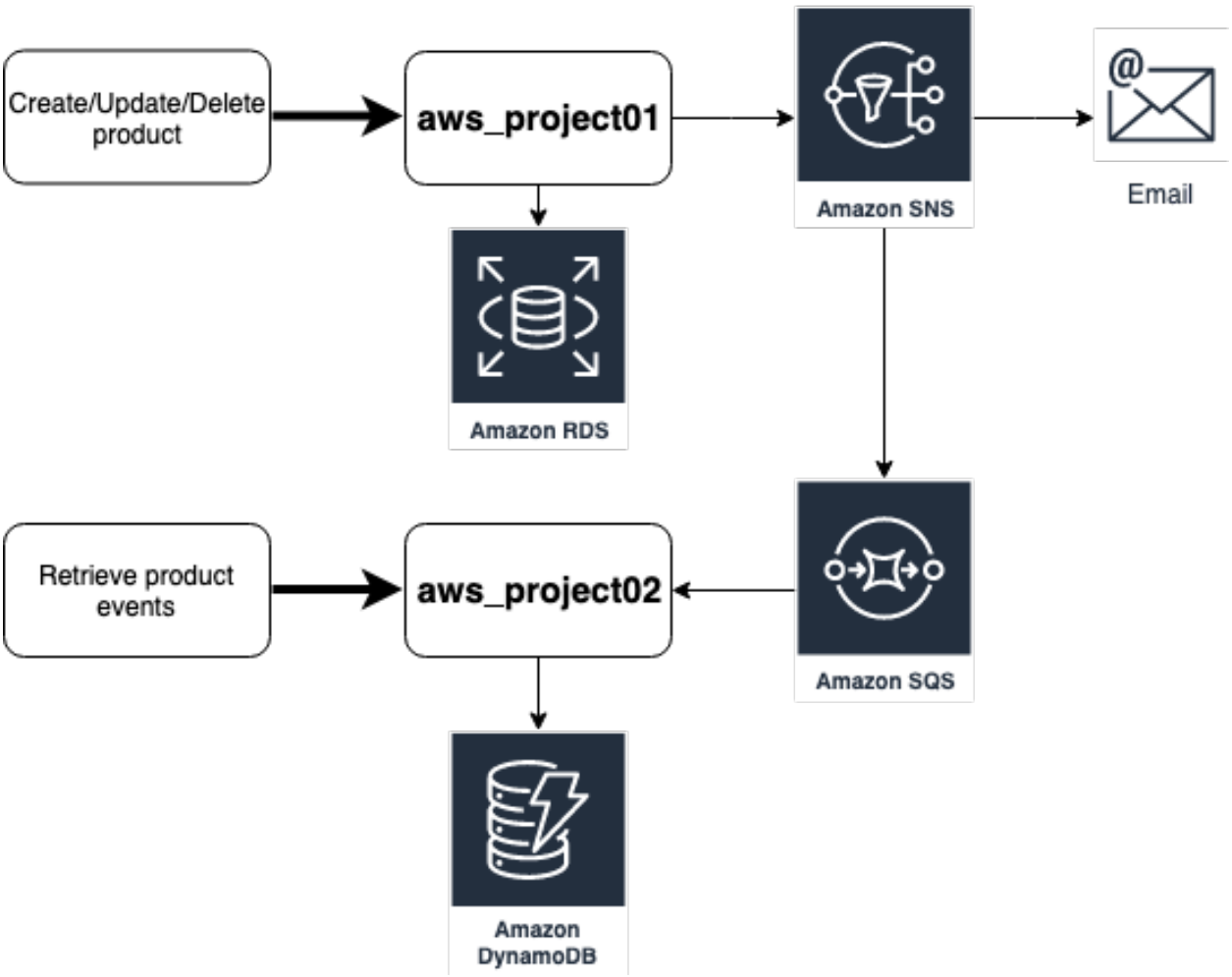
These topics can deliver messages to email addresses or can be used to post messages in queues, such as AWS **Simple Queue Service**, as will be presented in **chapter 13**:



SNS and SQS integration

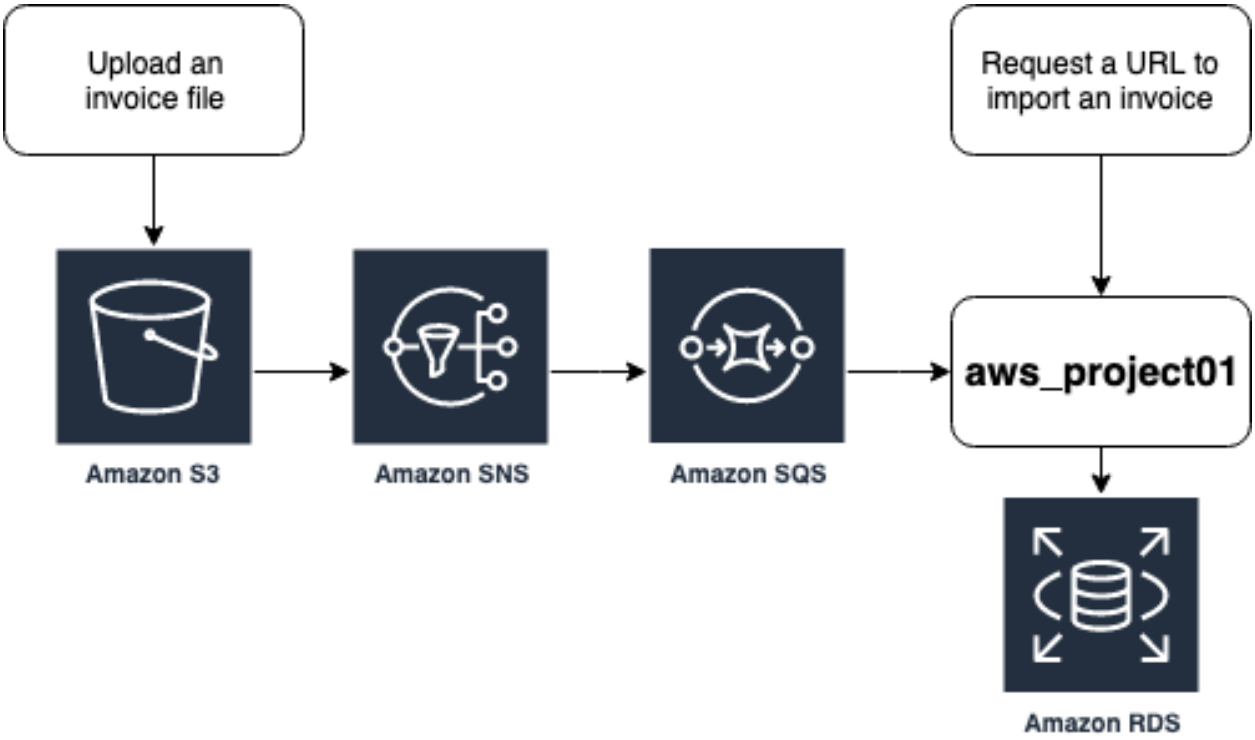
This way, communication can be created **asynchronously** between two applications.

Chapter 14 will detail basic concepts for **AWS DynamoDB**, a nonrelational database service, that allows applications to create items that can be stored in key-value as well as in documents.



Persisting events in the DynamoDB

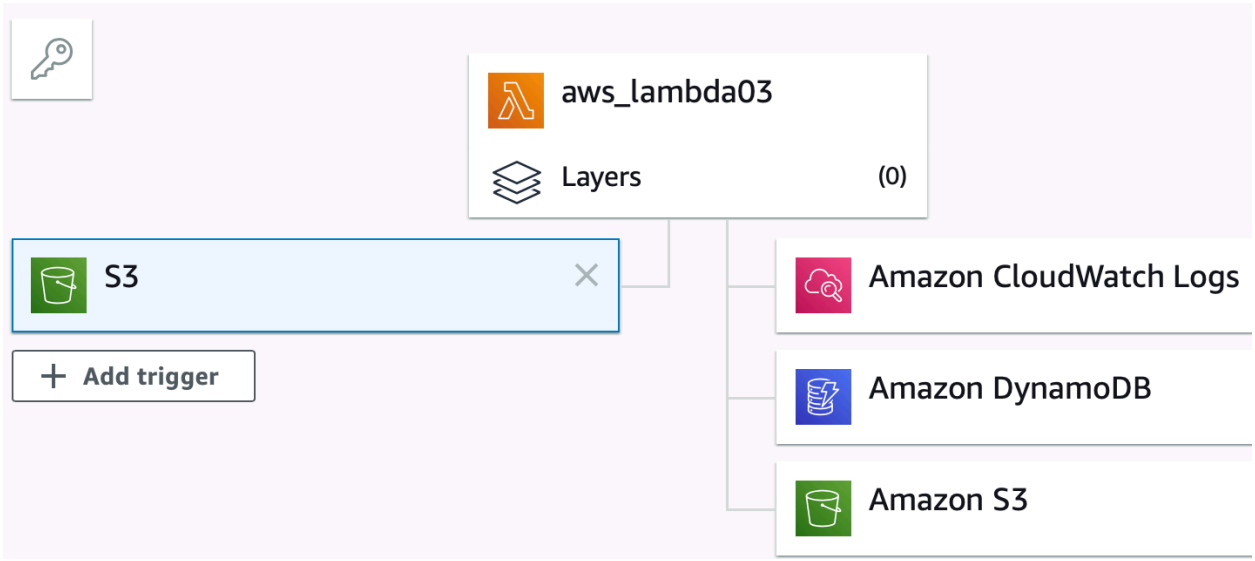
Chapter 15 will show how to use AWS **Simple Storage** Service for file storage, which can be managed through applications running in **clusters**:



Importing files with S3

The example to be developed is the integration between S3, SNS, SQS, ECS and RDS services, showing how an application can be notified of the insertion of a new file in an **S3 bucket**.

Finally, **Chapter 16** will introduce concepts of how to build serverless applications with AWS **Lambda**.



Lambda function accessing S3 and DynamoDB

Some examples will be developed to teach how Lambdas functions can be used for a wide range of applications. The previous figure shows a diagram of a Lambda function accessing an S3 bucket and persisting the file content to a DynamoDB table, as well as generating logs of its execution in CloudWatch Logs.

Enjoy this journey of knowledge and, if you need help, do not hesitate to call the author: siecola@gmail.com

Have fun!