



# Technical Researcher Intern

Summer Internship Report

Amazon Web Services Spain

Pedro Duarte da Costa

Master in Informatics and Computing Engineering

September 19, 2018

# 1 Introduction

My summer internship took place between June 16 and September 7, I was integrated with the Solutions Architect team at Amazon Web Services Spain. The team covers clients from the Spain, Portugal and Italy.

Throughout the internship, the sharing of experiences and group work have provided me with a remarkable growth and enrichment, both personally and professionally.

The work achieved throughout the internship will be presented in this document.

## 1.1 Motivation

Amazon Web Services, AWS, is the worldwide leader in cloud computing services. For this reason, I decided to apply for the position of Solutions Architect Intern, so that I could firstly know the full range of services and how to help clients migrate, develop and architect their services in the AWS cloud. Creating a unique opportunity to learn how to architecture for the cloud on a large scale.

## 1.2 Goals

I integrated the AWS Iberia's Solutions Architect team to help the team find ways to automate routine processes in which they spent a lot of time throughout the year when they could be dedicating that same time to their clients. At the same time, learning about customer needs, learning to fit into a culture of inter-help, knowledge, self-learning, high standards, and above all obsession with customers.

# 2 Achieved Results

- AWS Certified Solutions Architect - Associate;
- General knowledge of all AWS services;
- Practical knowledge of the following services: EC2, DynamoDB, RDS, S3, Cognito, SES, Lambda, Step Functions, Gateway API, CloudWatch, CloudWatch Events, among others;
- Experience integrating and architecting various services in order to achieve a scalable, secure and cost-efficient solution;
- Experience building completely serverless applications;
- Deepened my knowledge of software development skills using python and javascript;
- Improved knowledge on Chalice (python framework for serverless development) and Vue.js;
- Hands-on experience working on a team of Solutions Architects;

## 3 Work Performed

### 3.1 Customer Proficiency Plan (CPP)

#### 3.1.1 Contextualization

The team has developed a series of programmatic activities targeted to better drive the adoption of AWS solutions and generate demand, the Solutions Architect (SA) team has created the Customer Proficiency Plan as a customer enablement initiative. The Customer Proficiency Plan seeks to foster AWS adoption amongst customers through an enablement plan in the form of a well-known schedule of hands-on workshops. This involves Solutions Architects who deliver the workshops and Account Managers who in the end talk to customers in order to continue their relationship with AWS. As such the SAs need to provide accounts for customers to use.

#### 3.1.2 Impact

The CPP plan for the first half of 2018 comprises of 5 sessions (Intro to AWS, Data Analytics, DevOps, AI and SAP) and enabled 109 assistants among 75 different customers with a total of 4.37 customer satisfaction score and had a huge revenue influence.

#### 3.1.3 Challenge

We needed to understand the organizational challenges of delivering CPP workshops and develop an automation plan that was crystallized in a tool that delivers a one stop platform for Solutions Architects and Account Managers to, as automated as possible, create, manage and deliver the workshops to customers.

#### 3.1.4 Solution - First Stage

We needed to speed up the time taken to clean the accounts, these accounts are part of an organization and once created AWS does not allow them to be easily deleted. As such SAs spent a lot of time cleaning up the accounts for the next workshop. Given that the process involves entering each account, go through all AWS regions (16 regions) and each AWS service ( 100 services) and clean up the various components created we can see that it is a very time consuming and routine work.

I started by modifying a python script that spanned multiple accounts and regions and cleaned a few services. I began by redesigning the script, from a single file to a modular solution in which each service had an associated file. These files were automatically imported and their functions called. This way it became much easier to clean up new services because you just need to copy a service template file, develop your cleaning code for that service and put it in the folder.

Achieving the following structure:

```
1 CPP-Cleaner
2   |- services
3   |   |- __init__.py
4   |   |- service_template.py
5   |   |- ec2.py
6   |   |- dynamodb.py
7   |   |- ...
```

```
8 | - globals.py
9 | - init.py
10 | - main.py
```

The file `__init__.py` imports all services within the "services" folder and in `main.py` we get all the accounts within the organization of the master account and we go through all the accounts and regions within the accounts and call the cleaning functions in the "services" folder.

The service `_template.py` file serves as an example to add new services to clean.

In addition to simplifying the structure of this application, I added 10 more services to it.

This was the work of the first two weeks. With this, the SAs could already optimize their time after the workshop.

### 3.1.5 Solution - Second Stage

The next challenge was scaling the workshops and automate the process to deliver to a larger set of customers.

We began to review the requirements and objectives of the CPP. The main objective will always be to increase the business impact and customer satisfaction and to do so we had to scale the CPP for more workshops in order to reach more customers.

We raised several requirements in multiple meetings and set the goals for the first version of a web application that allows organizing the workshops, automatically notify Account Managers to invite their clients and allow customers to register for the workshop and get all the necessary information.

The manual CPP workflow and automated workflow can be found in Appendix A and B, accordingly.

The workshop then goes through several stages. It begins as a draft that defines the contents and context of the same, opening for inscriptions with unique links to invite customers, opening the workshop and automatically generate account credentials for the clients, alerts by email, closure of the workshop in which automatically the generated credentials are cleaned and the cleaner previously mentioned runs on an EC2 instance and clears all workshop accounts and automatically terminating itself at the end. As well as anonymizing the customers' details to comply with GDPR (General Data Protection Regulation).

The ultimate goal is to automate all possible processes so that we can offer the largest number of workshops per month by incurring in less time spent organizing everything for the SAs and Account Managers (AMs) and a greater benefit for customers.

As the only developer of the project, I was in responsible for defining the best architecture and developing it. I opted for a completely serverless architecture. With this, I abstained from maintenance and scaling of the infrastructure and focusing on the development.

The new architecture is in Appendix C.

This architecture comprises 4 components/repositories:

- **CPP Cleaner** - Can be run locally or through the tool to clean all the

accounts inside the organization and generate/delete clients' access credentials.

- **CPP Frontend** - Vue.js static interface for AMs and Tech team to manage almost all activities associated with the delivery of a CPP workshop.
- **CPP Rest API** - Built with chalice, supports the frontend by making all necessary updates and computational logic.
- **CPP Internal API** - Support API to send and schedule the sending of e-mails.

User authentication uses the AWS Cognito service, a serverless authentication service. I implemented passwordless authentication, so customers did not need to memorize another password/account, making the whole process seamless for everyone involved.

### 3.1.6 Outcome

A serverless internal tool that automates and scales the capture of new customers to the AWS platform and empowers existing customers. We hope to scale it to be used throughout AWS worldwide. I am very satisfied with the result achieved, I documented all the architecture and deployment of the tool and presented the project to the team of SAs who formed an internal team to continue the development.

## 3.2 CPP Workshop in Barcelona

I accompanied two SAs to Barcelona, in order to help in any doubt that the clients had during the workshop. In this activity, I had close contact with 3 specific clients with whom I tried to help in their doubts regarding the implementation of IoT, Serverless Solutions and Blockchain services in AWS. It proved a rewarding challenge, I had to respond in real-time to their questions, and then witness their enthusiasm in seeing how AWS services could help them improve their business.

## 3.3 Reusable Content and Materials

The team also keeps a repository of content they can all use to deliver presentations and workshops to customers. I helped improve this content by adding to presentations.

The first on **Cloud Computing with AWS** in which I explore the following topics:

- AWS Compute Business
- AWS Global Infrastructure concepts
- AWS Compute building blocks
  - Elastic Compute Cloud (EC2)
  - Managed Container Services
    - \* Amazon Elastic Container Service (ECS)
    - \* AWS Fargate

- \* Amazon Elastic Container Service for Kubernetes (EKS)
- \* Amazon Elastic Container Registry (ECR)
- Serverless compute: AWS Lambda
- AWS Batch
- Amazon Lightsail
- VMWare Cloud on AWS overview

The second presentation was on **Internet of Things (IoT)** describing the following services:

- AWS IoT Core
- AWS Greengrass
- Amazon FreeRTOS
- AWS IoT Analytics

### 3.4 AWS Certified Solutions Architect - Associate

During the internship, AWS provided me with several tools to certify as an AWS Certified Solutions Architect - Associate. As a member of the team, I had access to the complete course in [acloud.guru](http://acloud.guru) and to various internal resources and workshops that culminated in me passing the exam. The exam is described in AWS website as: "This exam validates an examinee's ability to effectively demonstrate knowledge of how to architect and deploy secure and robust applications on AWS technologies.". I learned in a high-level view of all AWS services and how to integrate them to build a secure, scalable and highly available architecture. Aside from theoretical knowledge I acquired practical knowledge in services such as Elastic Cloud Computing (EC2), Relational (RDS) and Non-Relational Databases (DynamoDB), Virtual Private Cloud, Serverless (Lambda, API Gateway , Cognito, among others) and Storage.

## 4 Conclusion

This internship provided me in 3 months, the equivalent of one semester of several courses. I learned how in practice AWS establishes itself as a market leader through customer obsession and their needs. I have developed soft skills working in a team, presented the results and project to the team and directed discussion with clients about AWS cloud architecture-related issues. At a technical level, direct contact with real customer stories, their difficulties and the corresponding solutions put me in a position of enormous learning in large-scale problems. With the CPP project, I learned how to create and develop a serverless architecture using the various AWS services. I also had access to internal workshops and internal content where I was able to learn immensely about new services, business perspectives, and success stories. Certification as a Solutions Architect complemented and validated all knowledge learned throughout the internship.

## Acknowledgments

Thanks to the entire AWS Iberia Solutions Architect team, they all welcomed me incredibly well and helped me with any challenge. Especially to my manager Carlos Carus, who has an incredible vision and knows how to pass knowledge and ideas always in a fun way. Also, special thanks to Nacho Garcia Alonso, Solution Architect, for working with me on the CPP project, always being ready to help and raise my interest in Serverless services and architectures.

## Contacts

**Tutor of Internship (FEUP)**

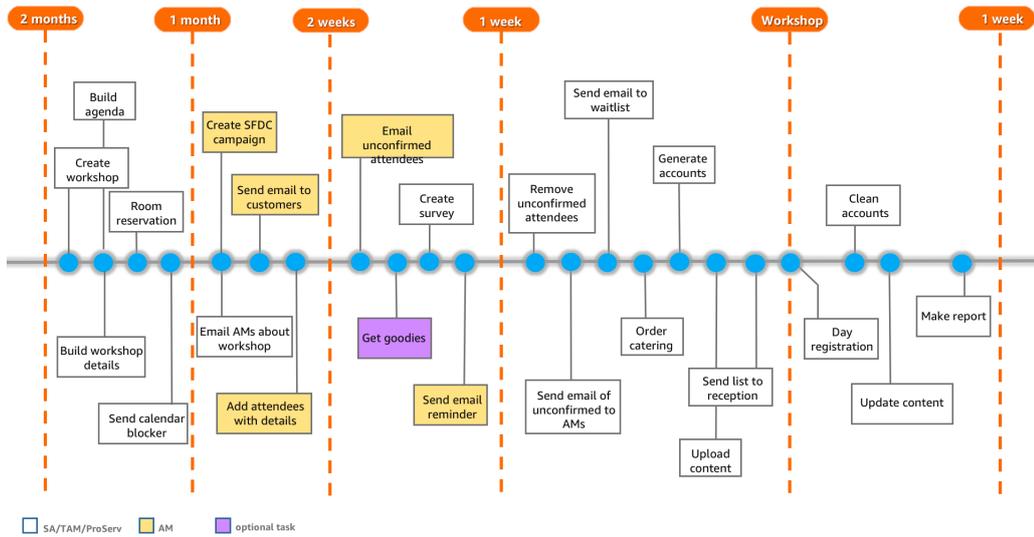
Tiago Boldt Sousa, tbs@fe.up.pt

**Tutor of Internship (Amazon Web Services)**

Carlos Carus, Head of Iberia Solutions Architecture, carusc@amazon.es

# Appendix

## A Initial CPP Workflow



## B New Automated Proposed CPP Workflow

