AWS Cert Alert
Functional Spec & Plan

Oisín Chelmiah
C00246756

Supervisor:
Dr Keara Barrett

Abstract

This document overviews some key aspects of the project, including what software is required for the system to run, the specification of the system, the users and use cases for the system, and how I plan on measuring the system's success. A project plan is also included outlining a timeline for the progression of the project.

Table of Contents

Table of Figures

Introduction

The main aim of this document is to describe how my system – AWS Cert Alert – will function. AWS Cert Alert is a system that can be implemented by any company that using Amazon Web Services (AWS) accounts to manage their Transport Layer Security (TLS) certificates. It is designed to aid in certificate management by keeping track of when TLS certificates expire and then providing a logging functionality and a notification system which will be connected to a dashboard. The system is designed entirely native to AWS account which allows it to be compatible with any business that has an AWS account. By designing the system entirely within AWS, it also allows for users to customise their experience by tailoring the system to suit their certificate management needs.

To show how adaptable my system is, the system is also going to monitor resources being used by Amazon Cloud Formation Stacks. The dashboard will detail what resources are being used and if any are currently not in use, as well as highlighting orphaned resources. Orphaned resources when an object/system/database that was using a resource gets deleted, but the resource itself remains.

Overview

Requirements

The system is designed using native AWS tools, so the main requirement is an AWS account. Within this account, the following services will need to be activated in order for the system to be created:

Service	Requirement
Certificate Manager (ACM)	Needed to store the certificates which will be processed.
Lambda	Used as a repository to store the code that will be used to design the system.
Step Functions	Used to create a State Machine which will be used to execute the processing functionality of the system.
Security Hub (SH)	Used to log vulnerabilities, in the case it will keep a log of certificates that are about to expire or have already expired.
Dynamo Database (DB)	Used to store certificate details once they have been retrieved from ACM.
Cloud Watch	Schedules how often the system will run.
Quick Sight Dashboard	Needed to create the display that will be used to show certificate details in an easy-to-read format.
Simple Notification Service (SNS)	Used for sending notifications from the system.
Identity & Access Management (IAM)	Needed to give the system permissions within the AWS account.
Cloud Formation	Used to create stacks that will be assigned resources.

Specification

The main functionality provided by the system is the tracking of when TLS certificates that are being stored within an AWS account will expire. The certificates in questions will be stored in AWS Certificate Manager. From here, the Cert Alert system will begin to process this system through the use of Lambda functions. These functions will integrate with a database to provide the logging functionality, SNS to send notifications, and Quick Sight to create the dashboard.

Logging	
Logging to Database – Core	Provide a historical record of all TLS certificate details in a database stored on Dynamo DB. Database entries will be made automatically by the system based on certificates stored in ACM.
Logging to Security Hub – Non-Core	Create a vulnerability warning in Security Hub if a certificate is coming up to its expiry date, if it has already expired, or if a resource has become orphaned.
Notification	
Email – Core	An email notification to a designated user notifying them the dashboard has been updated.
Text Message – Non-Core	Same as the email notification but in text format.
Dashboard	
Displaying Cert Details – Core	The main functionality of the dashboard is to display the details of stored TLS certificates in an easy-to-read format as well as highlight certificates that are about to expire or have already expired.
Analyse Signing Algorithms Used – Non-Core	An additional feature as part of the dashboard to analyse what signing algorithms each certificate is using and make recommendations for each.
Displaying Resources – Core	The dashboard will also have a separate page for displaying resources in use by Cloud Formation stacks, detailing what stacks are using what resources, whether any resources are currently not in use, and highlighting any orphaned resources.
Filtering – Non-Core	A filtering function to apply to the dashboard to allow viewers to view a more specific set of results.
Exporting – Non-Core	A functionality to take a snapshot of the dashboard and save it locally.
Sharing – Non-Core	An option to share the dashboard to external users. This would not provide those users access to the AWS account that the dashboard is connected to, only the dashboard itself.

Table 1 – Specification

Users

AWS Cert Alert is targeted at companies using AWS to manage their web facing applications. Depending on the size of the company implementing the AWS, the specific users of the system will vary.

For small companies and inidividuals with one AWS account the system will more than likely be used by the owner of the AWS account. For larger companies with multiple AWS accounts, certificate management may a part of the duties of an IAM team, a cloud security team, or a certificate management team. In this instance, the upkeep of the system would be the responsibility of whichever employee/team is in charge of certificate management.

The users of the dashboard are not limited to the users of the rest of the Cert Alert system. The dashboard designed in a way that is easy for the managers of the system to present its output to higher ups, such as executives or C-suite level managers, or to be used in reports for auditing. In this sense, while the scope of users of the entire system is narrowed down to whoever is in charge of certificate management, the scope of users of the dashboard is more broad.

Use Cases

Use Case Diagram

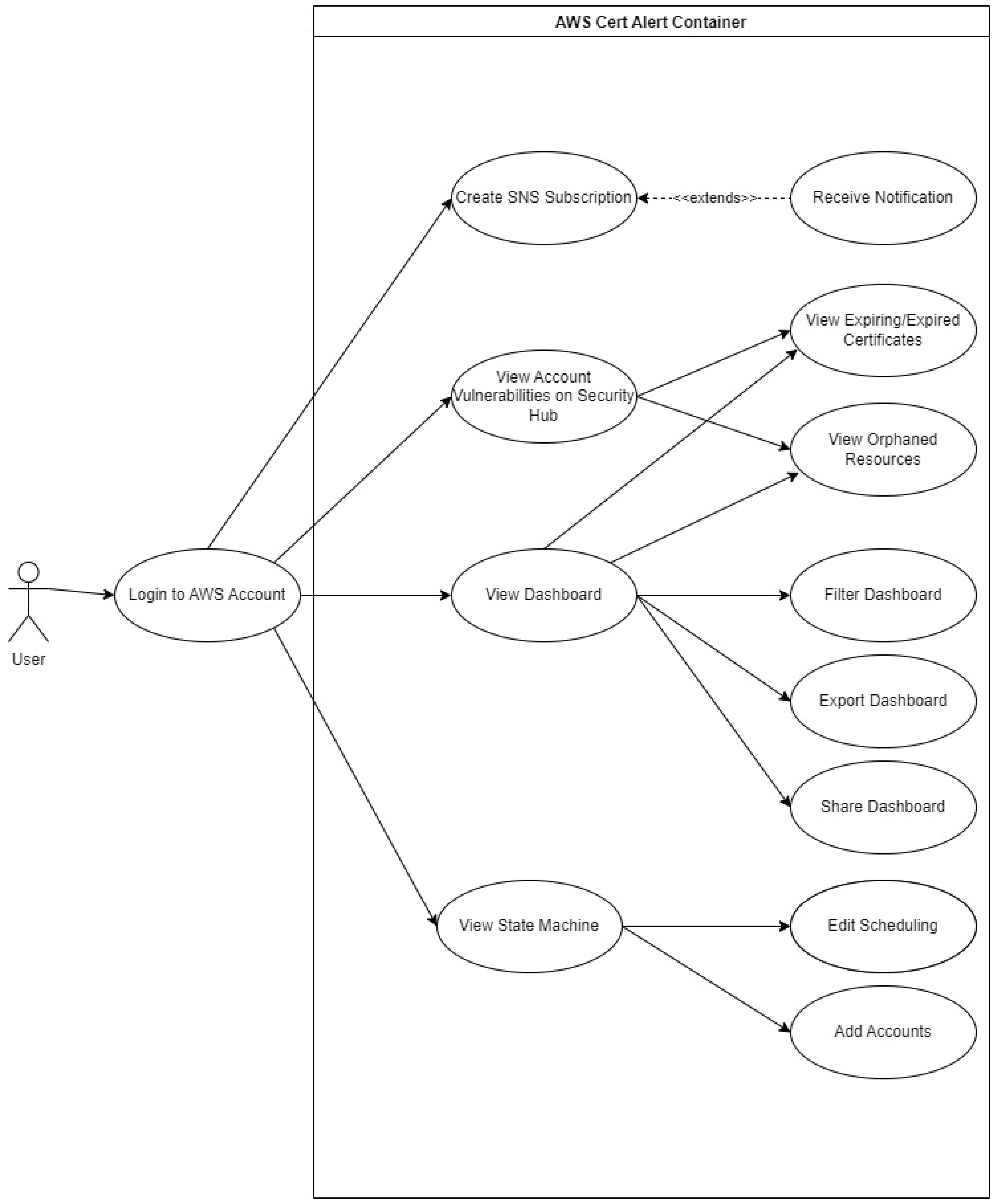


Figure 1 – Use Case Diagram

Use Case Description

No.	Component	Precondition	Steps	Output
1	Login to AWS Account	An AWS account.	-Navigate to the AWS login webpage. -Enter login details.	User can now access the AWS Cert Alert system.
2	Create SNS Subscription	1	-Navigate to the SNS service within the AWS account. -Create a new SNS topic. -Name the SNS topic. -Assign email addresses/phone numbers to the SNS topic. -Navigate to the Lambda service. -Click on the SendNotification function -Set the SNSTopic variable to be the name previously created SNS topic.	Email addresses/phone numbers subscribed to the SNS topic will be able to receive Cert Alert notifications.
3	Receive Notification	2	-View notification.	Relevant party has been notified that a certificate has an upcoming expiry date/has expired.
4	View Dashboard	1	-Navigate to Quick Sight Dashboard. -Click the AWS Cert Alert dashboard. -View the dashboard.	The dashboard provides data on the TLS certificates and stack resources in an easy-to-read format.
5	Filter Dashboard	1, 4	-Click the filter drop down menu. -Select the one of the filtering options.	The dashboard can be filtered to show more specific data.
6	Export Dashboard	1, 4	-Click the Export button. -Select the export location. -Select the file type you wish to save the export as.	Allows a snapshot of the dashboard to be taken and saved locally.
7	Share Dashboard	1, 4	-Click the Share button. -Type in the email address of who you want to share the dashboard to.	External users can now view the dashboard without direct access to the AWS account.
8	View Account Vulnerabilities on Security Hub	1	-Navigate to the Security Hub service. -Click View Vulnerabilities.	Logs an expiring/expired TLS certificate or an orphaned resource as a vulnerability.

9	View Expiring/Expired Certificates	1, 4/8	-Navigate to the AWS Cert Alert dashboard or the vulnerability entry for an expiring/expired certificate. -Click on the certificate ARN to view the certificate in ACM.	Users can navigate from either Security Hub or the Cert Alert dashboard to ACM.
10	View Orphaned Resources	1, 4/8	-Navigate to the AWS Cert Alert dashboard or the vulnerability entry for an expiring/expired certificate. -Click on the resource ARN to view the certificate in its home service.	Users can navigate from either Security Hub or the Cert Alert dashboard to view resources directly.
11	View State Machine	1	-Navigate to the Step Functions service. -Click on the CertAlert State Machine.	Users can view details of the State Machine, including what resources it uses and how often it runs.
12	Edit Lambdas	1, 11	-Click on the ARN of one of the Lambda functions in the State Machine. -Alternatively, navigate to the Lambda service and select a function to edit. -Edit the code as required.	Users can edit the code to suit their system needs if required.
13	Edit Scheduling	1, 11	-Click on the scheduling assistant in the State Machine details. -Change the schedule as required.	Users can customise how often they want the system to scan the certificates.
14	Add Accounts	1, 11	-Click Edit in the State Machine details. -Select the GetAccount function. -Click the Function drop down menu. -Select GetAccounts to implement multiple accounts instead of a single account.	Users can either scan a single AWS account or multiple accounts depending on their business size.

Table 2 – Use Cases

Design

Lambda Functions

Function	Input	Description	Output
1 – GetAccount (Can also be GetAccounts)	None	This is the very first function that will be run when the system is triggered. It scans the account it's in to get the AWS account ID as well of the IDs of any connected AWS accounts.	AWS account IDs.
2 – GetCerts	AWS account IDs.	This function uses the provided AWS account IDs to scan the certificates stored in the Certificate Manager service of those accounts.	TLS certificate details.
3 – GetResources	AWS account IDs.	Similar to the GetCerts function, this function will get the details of all the resources associated with Cloud Formation stacks.	Stack resource details.
4 – PushToDB	TLS certificate details and stack resource details.	This function simply takes in the details of the certificates/resources and logs them in a database stored on Dynamo DB. As it stores them it will keep a track of which certificates are going to expire soon or have already expired.	Expiring/Expired TLS certificates and orphaned resources.
5 – PushToDashboard	TLS certificate details and stack resource details.	Points the dashboard to the database so it can display the data in a reasonable format.	None
6 – PushToSH	Expiring/Expired TLS certificates and orphaned resources.	Logs a vulnerability in Security Hub containing the details of expiring/expired certificates and orphaned resources.	None
7 – SendNotification	Expiring/Expired TLS certificates.	Notifies a specified user that a certificate is coming up to its expiration date or a certificate has expired.	None

Table 3 – Lambda Functions

State Machine

AWS Step Functions allow users to create State Machines. The State Machine for the Cert Alert system is designed so that the output of the first function is the input for the next function, then the output of that function is the input for the next function, and so on. The State Machine is designed so that once all the TLS certificate details/resource details have been retrieved, it will enter into a for-each loop to process each certificate individually. This allows for modularity in my approach, meaning that if the processing of one certificate/resource fails or times out it will not affect the processing of the other certificates/resources. The State Machine also has an in built error detection system that will notify the account owner if any action fails during execution.

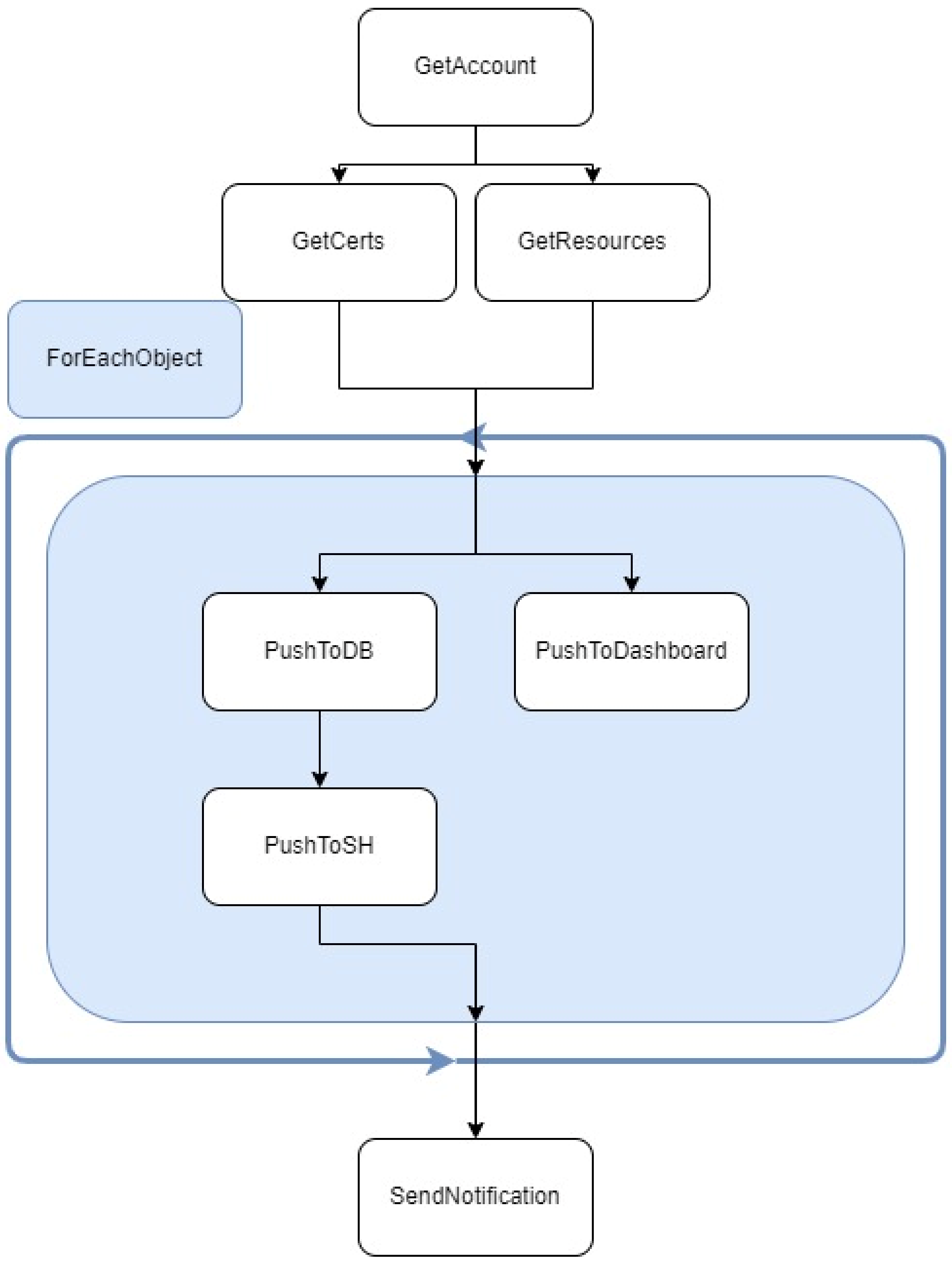


Figure 2 - State Machine

Notification

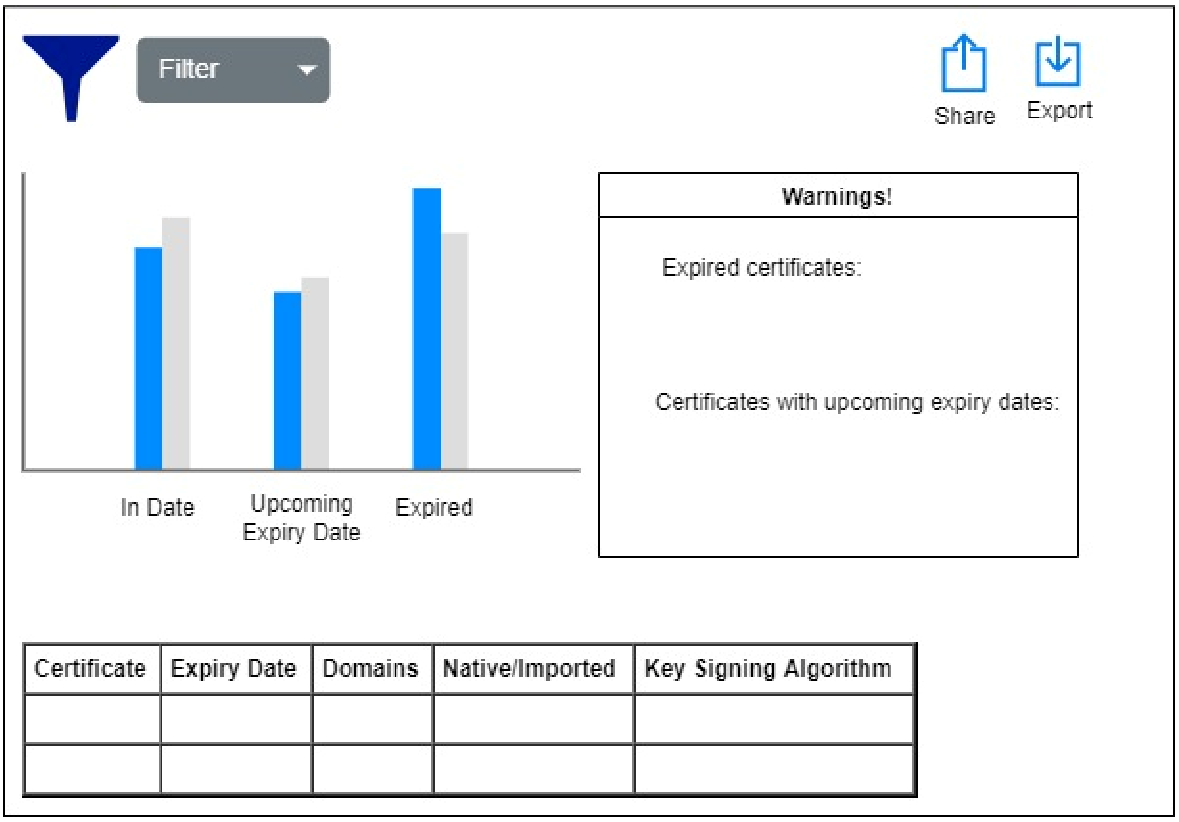
The main aim of the notification system is to direct users towards the dashboard and will be sent out using the SNS service. A link to the dashboard will be included in the email but it will only be accessible if the recipient is logged into the correct AWS account or if the dashboard was previously shared with them.

From:	AWS Cert Alert
To:	SNS Topic Subscribers
Subject:	AWS Cert Alert Notification
Body:	Your system currently has one or more TLS certificates with upcoming expiry dates. Please navigate to the AWS Cert Alert dashboard for more details.

Figure 3 – Sample Email Notification

Dashboard

The below is a mock up I designed of what the finished dashboard will look like when it is created from the certificate data stored in the connected AWS account.



Certificate	Expiry Date	Domains	Native/Imported	Key Signing Algorithm

Figure 4 – Mock Dashboard

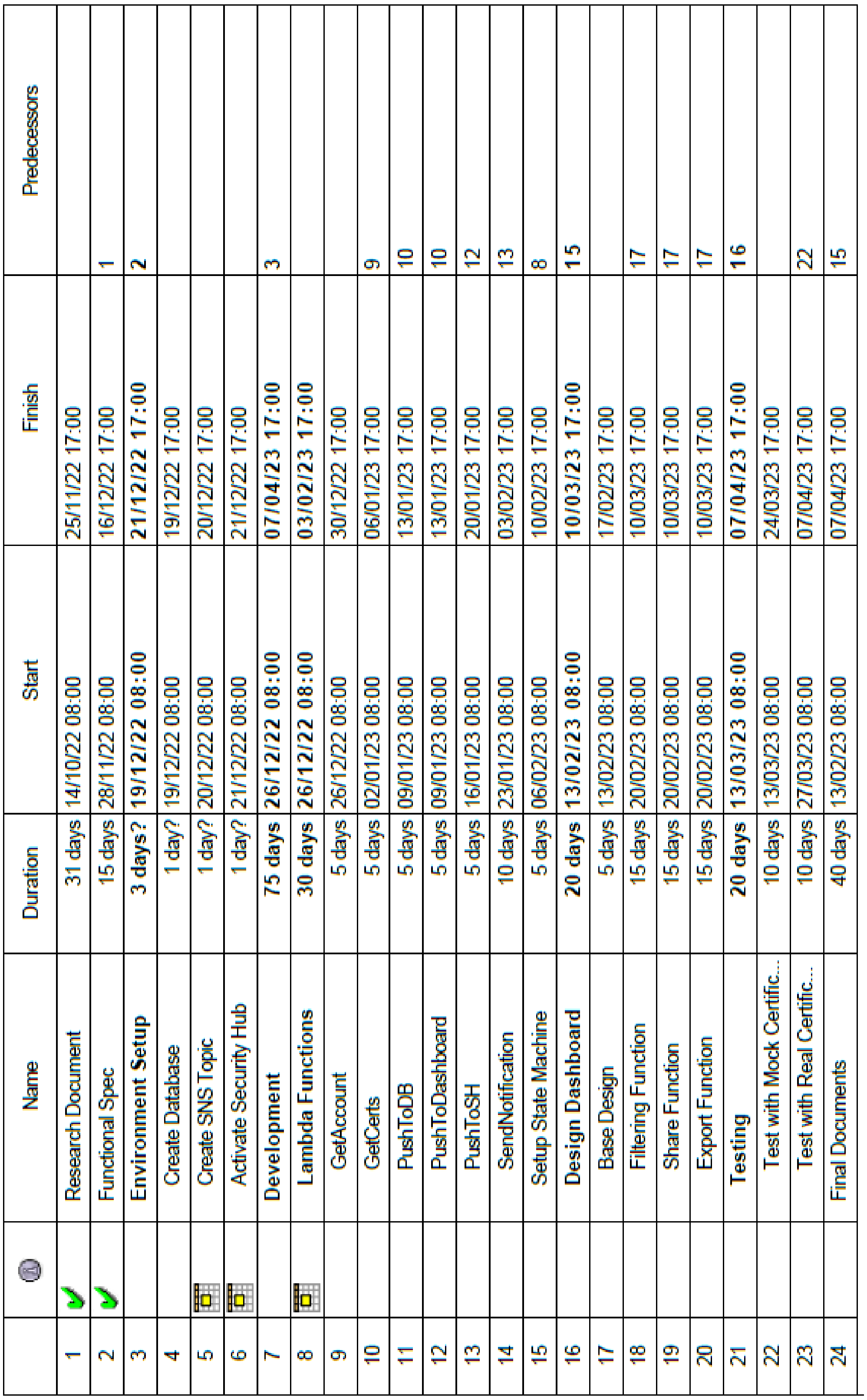
Testing

To test the success of this system, I’ve developed a three-stage testing process described below.

Phase One: Mock Certificates	AWS Cert Alert is designed for certificate management, but it would be impossible to test the system without any TLS certificates. For the first phase of testing, I’ve creating a Lambda function that will create an object mimicking a TLS certificate. This mock certificate will contain all the details that a regular certificate would without any of the functionality of an actual certificate. This Lambda function will be run instead of the GetCerts function and will output the details of the mock certificate. The mock certificate will be created in an array which allows me to create multiple mock certificates to test the performance of the system.
Phase Two: Single Account Certificates	The next step in testing is to use actual TLS certificates instead of mock certificates. These certificates will be stored in ACM within the same AWS account that the Cert Alert system is set up in. This means that the Cert Alert system will only be dealing with certificates on its local account. This will be used to test how the system performs in a small business that is only using one AWS account to control its certificates.
Phase Three: Multi-Account Certificates	The final stage in testing involves creating multiple AWS accounts under one business account. Each individual AWS account will have its own TLS certificates. The AWS Cert Alert system will either be set up in a new account with read access to the other accounts’ ACMs. This will test how the system would perform within a large-scale organisation.

Table 4 – Testing

Plan



	ⓘ	Name	Duration	Start	Finish	Predecessors
1	✓	Research Document	31 days	14/10/22 08:00	25/11/22 17:00	
2	✓	Functional Spec	15 days	28/11/22 08:00	16/12/22 17:00	1
3		**Environment Setup**	**3 days?**	**19/12/22 08:00**	**21/12/22 17:00**	**2**
4		Create Database	1 day?	19/12/22 08:00	19/12/22 17:00	
5	▦	Create SNS Topic	1 day?	20/12/22 08:00	20/12/22 17:00	
6	▦	Activate Security Hub	1 day?	21/12/22 08:00	21/12/22 17:00	
7		**Development**	**75 days**	**26/12/22 08:00**	**07/04/23 17:00**	**3**
8	▦	**Lambda Functions**	**30 days**	**26/12/22 08:00**	**03/02/23 17:00**	
9		GetAccount	5 days	26/12/22 08:00	30/12/22 17:00	
10		GetCerts	5 days	02/01/23 08:00	06/01/23 17:00	9
11		PushToDB	5 days	09/01/23 08:00	13/01/23 17:00	10
12		PushToDashboard	5 days	09/01/23 08:00	13/01/23 17:00	10
13		PushToSH	5 days	16/01/23 08:00	20/01/23 17:00	12
14		SendNotification	10 days	23/01/23 08:00	03/02/23 17:00	13
15		Setup State Machine	5 days	06/02/23 08:00	10/02/23 17:00	8
16		**Design Dashboard**	**20 days**	**13/02/23 08:00**	**10/03/23 17:00**	**15**
17		Base Design	5 days	13/02/23 08:00	17/02/23 17:00	
18		Filtering Function	15 days	20/02/23 08:00	10/03/23 17:00	17
19		Share Function	15 days	20/02/23 08:00	10/03/23 17:00	17
20		Export Function	15 days	20/02/23 08:00	10/03/23 17:00	17
21		**Testing**	**20 days**	**13/03/23 08:00**	**07/04/23 17:00**	**16**
22		Test with Mock Certific...	10 days	13/03/23 08:00	24/03/23 17:00	
23		Test with Real Certific...	10 days	27/03/23 08:00	07/04/23 17:00	22
24		Final Documents	40 days	13/02/23 08:00	07/04/23 17:00	15

Table 5 – Project Plan

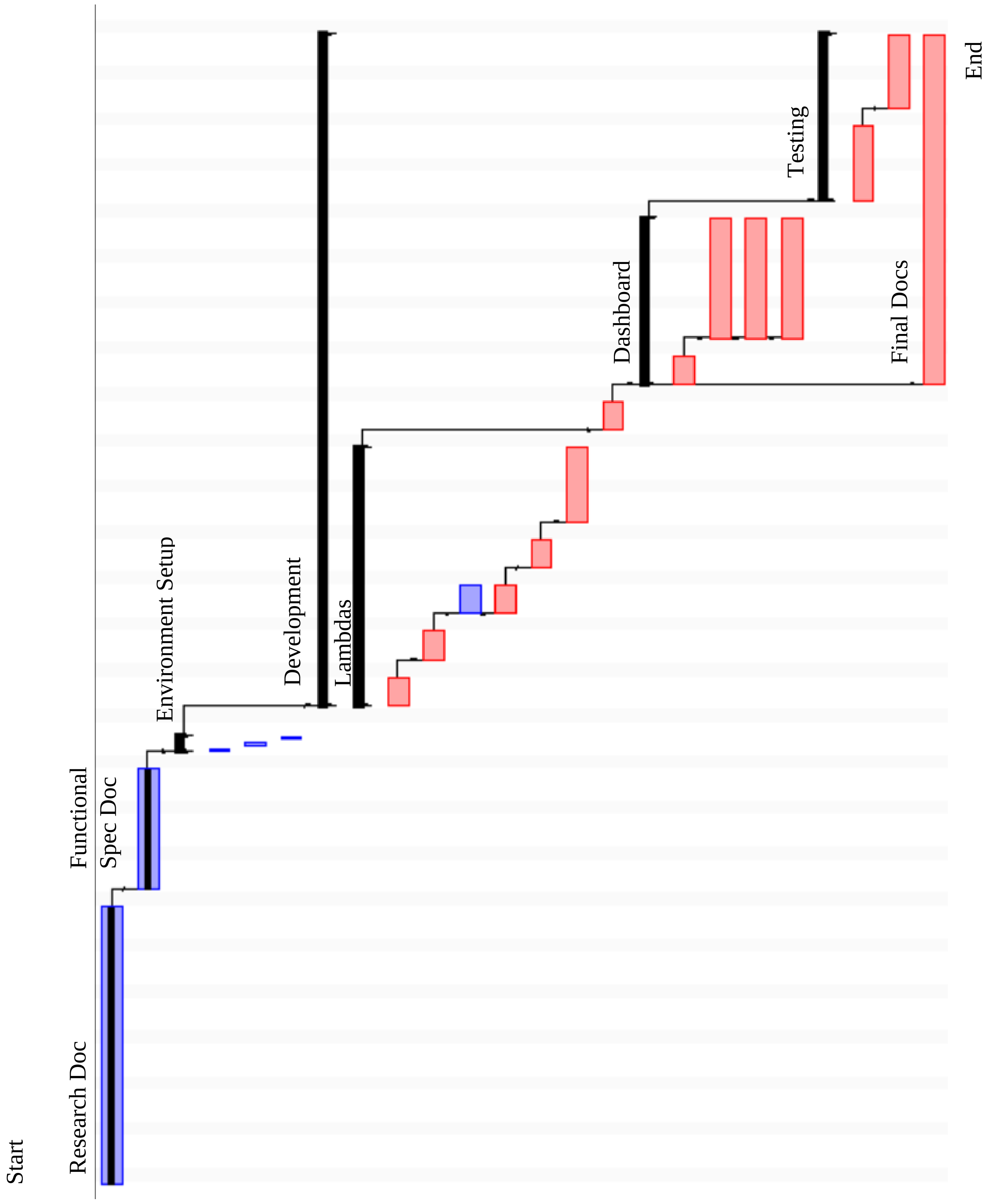


Figure 5 – Gantt Chart

Conclusion – Precedent for Application

TLS certificates are what enable websites to create secure connections with their clients. If a certificate were to expire, it could result in disastrous outcomes for both the website owners and the clients. Best case scenario: the website will deny the client access to its services. Worst case scenario: the website still provides the client with the desired service but their connection is now insecure, leaving both open to threat actors sniffing and altering their transactions.

There are certificate management applications out there, but any that currently exist are stand-alone third-party tools. This results in more charges, more user-manuals to learn, possible compatibility issues, and on top of that these services are not open source.

Being designed entirely native to AWS means that the Cert Alert system can be easily implemented into any system that is using AWS to manage its web-based services. The Cert Alert system also has the benefit of being compatible with any third-party tool that AWS is compatible with.

Another benefit of the Cert Alert is that it is very easily customisable by design. Users can change the scheduling of the system, customise who gets notified by the system, choose who can see the dashboard, and even edit the code if needs be since the system is open source.

The resource tracker functionality provides a novel approach to resource management. The implementation of a tracker to highlight orphaned resources is an integral part of the system, since orphaned resources often go unnoticed. The resource tracker part of the system is built off of the same architecture as the certificate management part, showcasing just how adaptable the system can be.

Glossary

AWS – Amazon Web Services

TLS – Transport Layer Security

ACM – Amazon Certificate Manager

SH – Security Hub

IAM – Identity & Access Management

DB – Database

SNS – Simple Notification Service

ARN – Amazon Resource Name