INTERNSHIP REPORT

FEB 2021 – JULY 2021

Internship report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

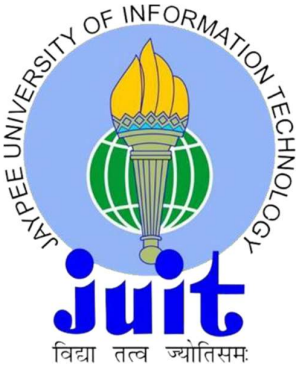
In

COMPUTER SCIENCE ENGINEERING

By:

Asmita Prajapati(171267)

To



Department of Computer Science & Engineering and Information Technology

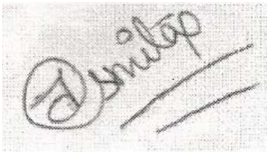
Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

TABLE OF CONTENTS

Chapter No.	Topics	Page No.
	Student Declaration	2
	Certificate from the Supervisor	3
	Acknowledgement	4
	Summary	5
	List of Figures	6
	Glossary of terms used	7
Chapter 1	Company's Profile	8-9
Chapter 2	Introduction to the Project	10
Chapter 3	Implementation details	11-36
Chapter 4	Results & Conclusion	37
	References	38

DECLARATION

I hereby declare that this submission is my own work carried out at **Watchguard Technologies India Pvt Ltd, Noida** from **Feb, 2021** to **July, 2021** and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma from a university or other institute of higher learning, except where due acknowledgment has been made in the text.



Signature

Name: **Asmita Prajapati**

Date: **22-05-2021**

CERTIFICATE

This is to certify that **Ms. Asmita Prajapati** of Jaypee University of Information Technology carried out the internship under my supervision at **Watchguard Technologies** from **Feb, 2021** to **July, 2021**. Her efforts in the development of this internship were satisfactory.

Arsh Arafaat

Engineering Manager

Watchguard Technologies

Date: 22 May, 2021

ACKNOWLEDGEMENT

I take this opportunity to express my sincere thanks and deep gratitude to all those people who extended their wholehearted cooperation and have helped me in completing this internship successfully.

First of all, I would like to thank Mr. Arsh Arafaat and Mr. Anand Dev, who mentored me, guided me and challenged me.

I also thank my family and friends who greatly supported me during the course of theInternship.

Last but not the least, I would like to thank our founders for considering me a part of the organization and provide such a great Platform to learn and enhance my skills.

A very special thanks goes to all the faculties of Jaypee University of Information Technologyunder whom guidance I have been able to excel in my career and become a part of the Watchguard family.

Asmita Prajapati

171267

Jaypee University of Information Technology

SUMMARY

This report is all about what I learned as an intern and the work I carried out in Watchguard Technologies, Noida during my internship period from Feb, 2021 to July, 2021.

For 25 years, WatchGuard has pioneered cutting-edge cybersecurity technology and delivered it as easy-to-deploy and easy-to-manage solutions. With industry leading network and endpoint security, secure Wi-Fi, multifactor authentication, and network intelligence products and services, WatchGuard enables more than 250,000 small and midsize enterprises from around the globe to protect their most important assets i ncluding over 10 million endpoints. In a world where the cybersecurity landscape is constantly evolving, and new threats emerge each day, WatchGuard makes enterprise
grade cybersecurity technology accessible for every company. WatchGuard is headquartered in Seattle, Washington, with offices throughout North America, Europe, Asia Pacific, and Latin America.

Working here has taught me that a project is not only a piece of code, it is a compilation of uncountable number of modules and a process behind building these modules. Writing code is just a small fraction of making an application. Planning, assigning, reviewing, fixing, testing, compiling and tracking all this process are some other fractions of developing an application.

During this internship, I did various Udemy trainings on, cyber-security, python, restful APIs in flask, postman and its testing. After that I have learning about cloud and its different services. I have also done two courses on AWS, explored various services with hands on experience in it. Currently, I am working on an asset-management project for the company using dynamodb, lambda, cognito, sns and cloudformation.

Asmita Prajapati
May 22, 2021

LIST OF FIGURES

1.	*IAM Policies*	15
2.	*Password Encryption*	16
3.	*S3 Buckets*	18
4.	*Cognito User Pool*	19
5.	*API Gateway*	21
6.	*Lambda Authorizer*	22
7.	Lambda Console	23
8.	*Dynamo DB table structure*	24
9.	*Cloudwatch Metric*	25

ABBREVIATIONS

1. **AWS -** Amazon Web Services
2. **EC2 -** Elastic Compute Cloud
3. **ECS -** Elastic Container Service
4. **EFS -** Elastic File System
5. **API -** Application Programming Interface
6. **RDS -** Relational Databases
7. **S3 -** Simple Storage Service
8. **KMS -** Key Management Service
9. **IAM –** Identity and Access Management
10. **ECR –** Elastic Container Registry

Chapter - 1

COMPANY'S PROFILE

1.1. Summary

Website:

https://www.watchguard.com

Facebook Page:

https://www.facebook.com/watchguardtechnolges

Linkedin Page:

https://www.linkedin.com/company/watchguard-technologies/

Twitter Page:

https://twitter.com/watchguard

1.2. About Us

For 25 years, WatchGuard has pioneered cutting-edge cybersecurity technology and delivered it as easy-to-deploy and easy-to-manage solutions. With industry leading network and endpoint security, secure Wi-Fi, multifactor authentication, and network intelligence products and services, WatchGuard enables more than 250,000 small and midsize enterprises from around the globe to protect their most important assets including over 10 million endpoints. In a world where the cybersecurity landscape is constantly evolving, and new threats emerge each day, WatchGuard makes enterprise grade cybersecurity technology accessible for every company. WatchGuard is headquartered in Seattle, Washington, with offices throughout North America, Europe, Asia Pacific, and Latin America.

Intelligent Protection

Effective protection against today's vast number of evolving threats requires multiple services working intelligently together. Prevent, detect, and instantly respond to cyber attacks with automated policies.

Simplified Management

Managing security across your organization has never been simpler. Use out-of-the-box tools to quickly and easily deploy, configure, and maintain your security with the granularity of your choice.

Actionable Visibility

Monitor and report on the health of your IT infrastructure. Actionable visibility tools enable you to proactively identify threats, while providing corrective action against known issues.

Chapter - 2

INTRODUCTION TO THE PROJECT

Asset Management project deals with the information of all the office related purchases of different items/assets including laptops, mice, routers, keyboards, monitors, etc. Managing their details, model, date of expiration and their availability is done in this project.

It uses different services of AWS for different purposes, from which the main ones are:

1) **DynamoDb**: For managing the serverless database of different kinds of assets
2) **Cognito**: For creating user pool and keep a track of different users who logs in
3) **Lambda Authorizer**: For authentication of various APIs and make it more secure
4) **Lambda**: For creating and storing our functional codes in a serverless manner
5) **API Gateway/Postman**: For testing our api and creating routes for different APIs
6) **Cloudwatch logs**: For maintaining logs for each lambda function we have created, and tracing its events
7) **S3**: For the static hosting of our websites
8) **Route 53**: For providing us with a domain name
9) **SAM template, Cloudformation**: For deploying it completely serverless

Chapter - 3

IMPLEMENTATION DETAILS

- **AWS and its services**

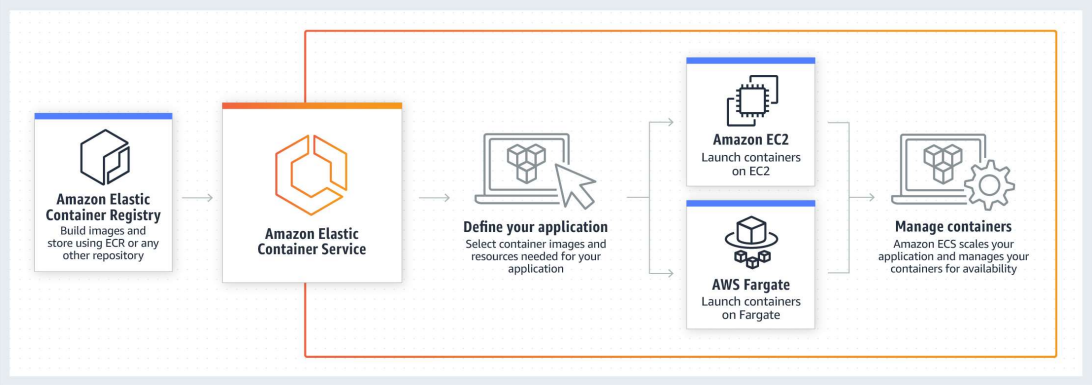
1. EC2

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resi zable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with co mplete control of your computing resources and lets you run on Amazon's proven computi ng environment.

Amazon EC2 offers the broadest and deepest compute platform with choice of processor, s torage, networking, operating system, and purchase model. We offer the fastest processors in the cloud and we are the only cloud with 400 Gbps ethernet networking. We have the m ost powerful GPU instances for machine learning training and graphics workloads, as well as the lowest cost-per-

inference instances in the cloud. More SAP, HPC, Machine Learning, and Windows workl oads run on AWS than any other cloud. We can connect to our ec2 servers using ec2 conn ect from the console itself or using some third party application like mobaxterm and putty server.

2. ECS / Fargate / ECR



What is ECS (Elastic Container Service)?

Amazon Elastic Container Service (Amazon ECS) is a highly scalable, high-performance container orchestration service that supports Docker containers and allows yo u to easily run and scale containerized applications on AWS. Amazon ECS eliminates the need for you to install and operate your own container orchestration software, manage and scale a cluster of virtual machines, or schedule containers on those virtual machines.

With simple API calls, you can launch and stop Docker-enabled applications, query the complete state of your application, and access many famili ar features such as IAM roles, security groups, load balancers, Amazon CloudWatch Event s, AWS CloudFormation templates, and AWS CloudTrail logs.

What is ECR (Elastic Container Registry)?

Amazon Elastic Container Registry (ECR) is a fully-managed Docker container registry that makes it easy for developers to store, manage, and deploy Docker container images. Amazon ECR is integrated with Amazon Elastic Container Service (ECS), simplifying your development to production workflow. Amazon ECR eliminates the need to operate your own container repositories or worry about scaling the underlying infrastructure. Amazon ECR hosts your images in a highly available and scalable architecture, allowing you to reliably deploy containers for your applications. Integration with AWS Identity and Access Management (IAM) provides resource-level control of each repository. With Amazon ECR, there are no upfront fees or commitments. You pay only for the amount of data you store in your repositories and data transferred to the Internet

What is AWS Fargate?

AWS Fargate is a compute engine for Amazon ECS that allows you to run containers with out having to manage servers or clusters. With AWS Fargate, you no longer have to provis ion, configure, and scale clusters of virtual machines to run containers. This removes the n eed to choose server types, decide when to scale your clusters, or optimize cluster packing. AWS Fargate removes the need for you to interact with or think about servers or clusters. Fargate lets you focus on designing and building your applications instead of managing th e infrastructure that runs them.

3. IAM

IAM is a feature of your AWS account offered at no additional charge. You will be charge d only for use of other AWS services by your users.

AWS IAM allows you to:

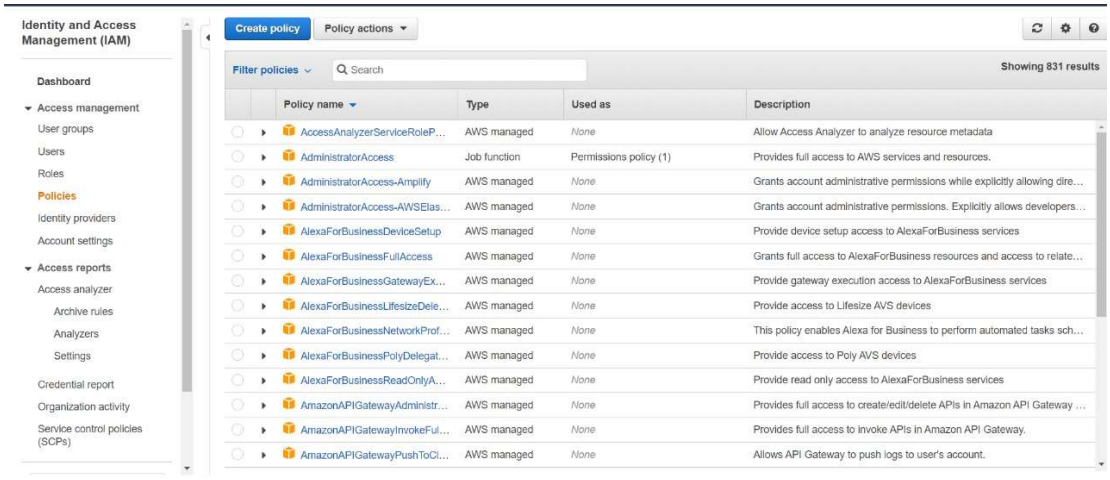
Manage IAM users and their access –
You can create users in IAM, assign them individual security credentials (in other words, access keys, passwords, and multi-
factor authentication devices), or request temporary security credentials to provide users ac cess to AWS services and resources. You can manage permissions in order to control whic h operations a user can perform.

Manage IAM roles and their permissions –

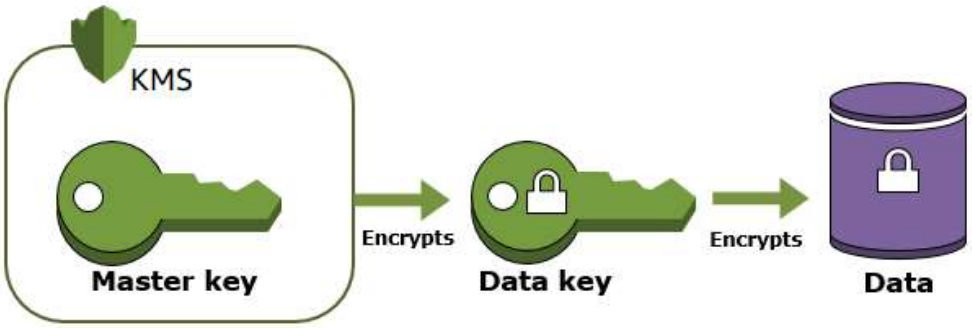
You can create roles in IAM and manage permissions to control which operations can be performed by the entity, or AWS service, that assumes the role. You can also define which entity is allowed to assume the role. In addition, you can use service-linked roles to delegate permissions to AWS services that create and manage AWS resources on your behalf.

Manage federated users and their permissions –

You can enable identity federation to allow existing identities (users, groups, and roles) in your enterprise to access the AWS Management Console, call AWS APIs, and access resources, without the need to create an IAM user for each identity. Use any identity management solution that supports SAML 2.0, or use one of our federation samples (AWS Console SSO or API federation).



4. KMS



AWS Key Management Service (AWS KMS) is an encryption and key management web s ervice.

Encrypts plaintext into ciphertext by using a customer master key (CMK). The Encrypt op eration has two primary use cases:

- You can encrypt small amounts of arbitrary data, such as a personal identifier or da tabase password, or other sensitive information.

- You can use the Encrypt operation to move encrypted data from one AWS Region to another. For example, in Region A, generate a data key and use the plaintext key to enc rypt your data. Then, in Region A, use the Encrypt operation to encrypt the plaintext data key under a CMK in Region B. Now, you can move the encrypted data and the encrypted data key to Region B. When necessary, you can decrypt the encrypted data key and the en crypted data entirely within in Region B.

Decrypts ciphertext that was encrypted by a AWS KMS customer master key (CMK) usin g any of the following operations:

Encrypt

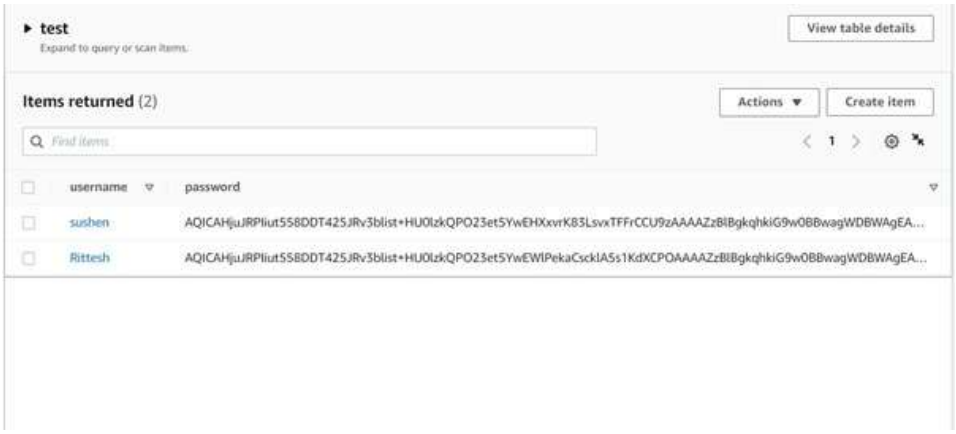
GenerateDataKey

GenerateDataKeyPair

GenerateDataKeyWithoutPlaintext

GenerateDataKeyPairWithoutPlaintext

You can use this operation to decrypt ciphertext that was encrypted under a symmetric or asymmetric CMK.

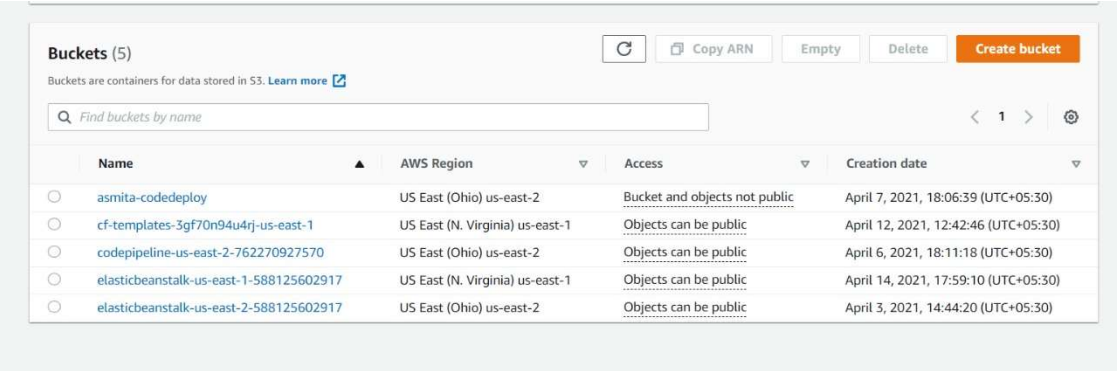


5. S3

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers indus try-

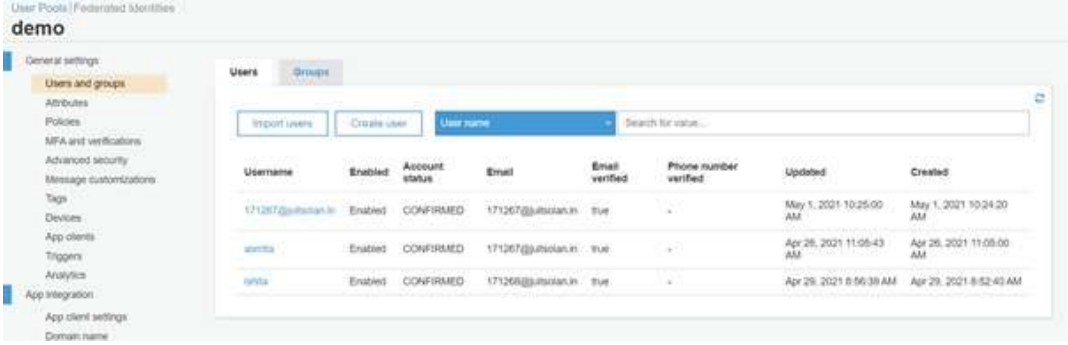
leading scalability, data availability, security, and performance. This means customers of a ll sizes and industries can use it to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enter prise applications, IoT devices, and big data analytics. Amazon S3 provides easy-to-use management features so you can organize your data and configure finely-tuned access controls to meet your specific business, organizational, and compliance requir ements.

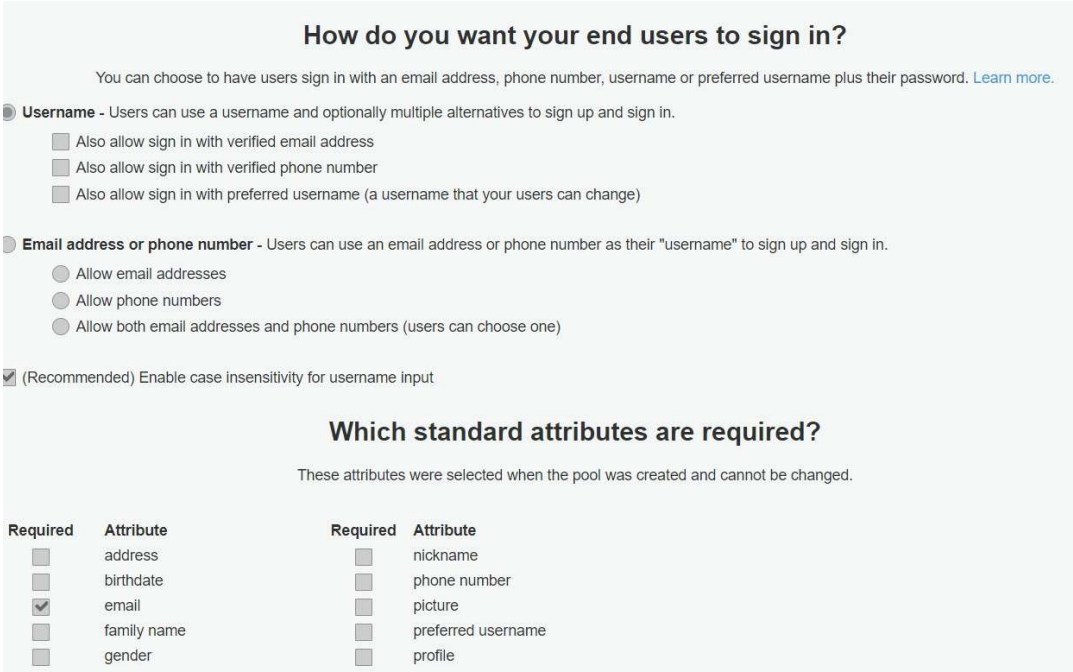
An Amazon S3 bucket is a public cloud storage resource available in Amazon Web Servic es' (AWS) Simple Storage Service (S3), an object storage offering. Amazon S3 buckets, w hich are similar to file folders, store objects, which consist of data and its descriptive meta data



6. Cognito

Amazon Cognito lets you add user sign-up, sign-
in, and access control to your web and mobile apps quickly and easily. Amazon Cognito s cales to millions of users and supports sign-
in with social identity providers, such as Apple, Facebook, Google, and Amazon, and ente rprise identity providers via SAML 2.0 and OpenID Connect.

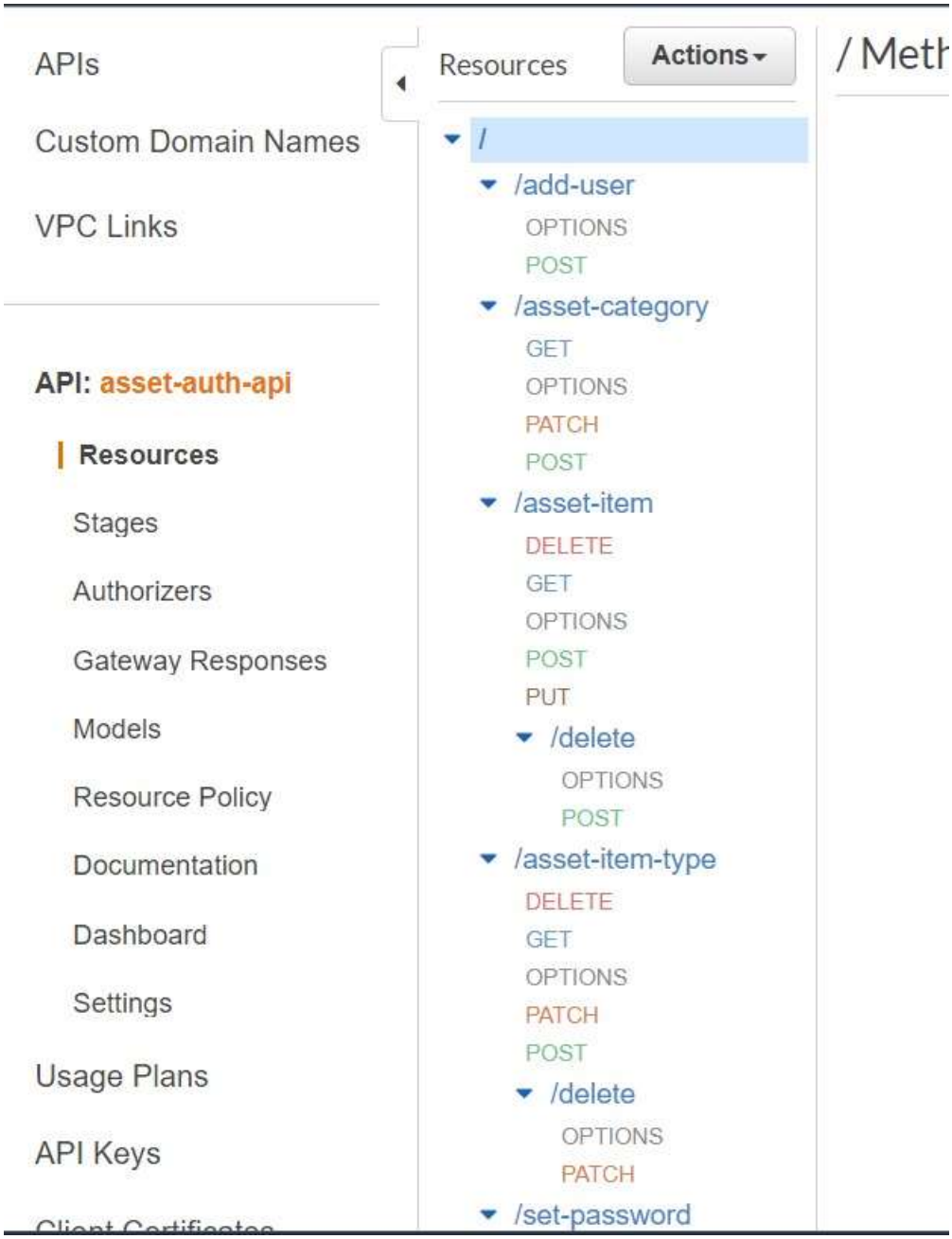




7. API Gateway

Amazon API Gateway is a fully managed service that makes it easy for developers to creat e, publish, maintain, monitor, and secure APIs at any scale. APIs act as the "front door" fo r applications to access data, business logic, or functionality from your backend services. Using API Gateway, you can create RESTful APIs and WebSocket APIs that enable real-time two-
way communication applications. API Gateway supports containerized and serverless wor kloads, as well as web applications.

API Gateway handles all the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls, including traffic management, CORS support, authoriza tion and access control, throttling, monitoring, and API version management. API Gatewa y has no minimum fees or startup costs. You pay for the API calls you receive and the am ount of data transferred out and, with the API Gateway tiered pricing model, you can redu ce your cost as your API usage scales.

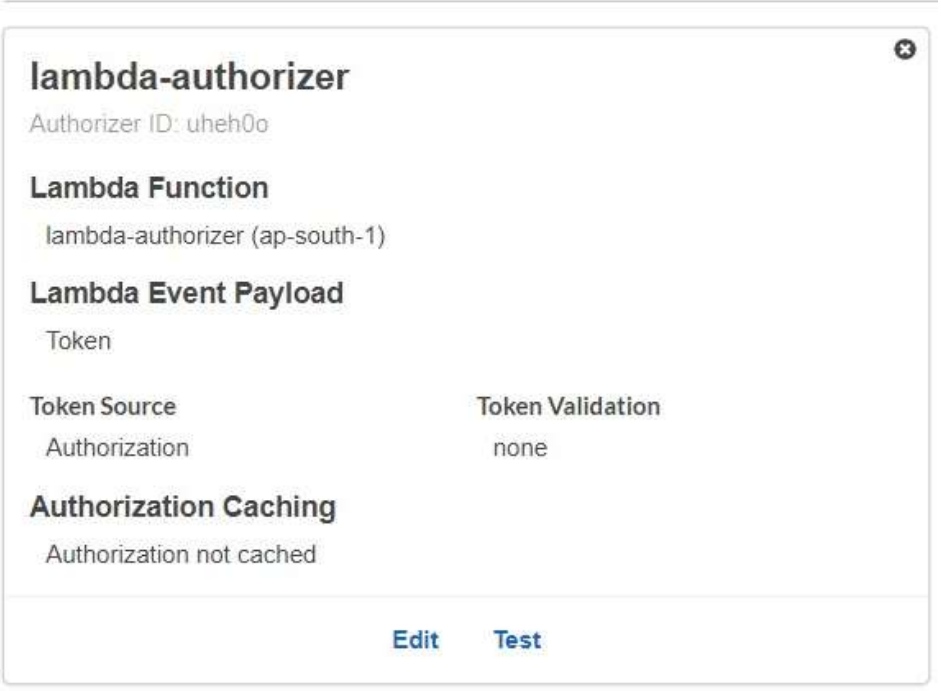

APIs
Custom Domain Names
VPC Links
API: asset-auth-api
Resources
Stages
Authorizers
Gateway Responses
Models
Resource Policy
Documentation
Dashboard
Settings
Usage Plans
API Keys
Resources
Actions
/
/add-user
OPTIONS
POST
/asset-category
GET
OPTIONS
PATCH
POST
/asset-item
DELETE
GET
OPTIONS
POST
PUT
/delete
OPTIONS
POST
/asset-item-type
DELETE
GET
OPTIONS
PATCH
POST
/delete
OPTIONS
PATCH
/set-password

Lambda Authorizer

A Lambda authorizer (formerly known as a custom authorizer) is an API Gateway feature that uses a Lambda function to control access to your API.

A Lambda authorizer is useful if you want to implement a custom authorization scheme th at uses a bearer token authentication strategy such as OAuth or SAML, or that uses reques t parameters to determine the caller's identity.

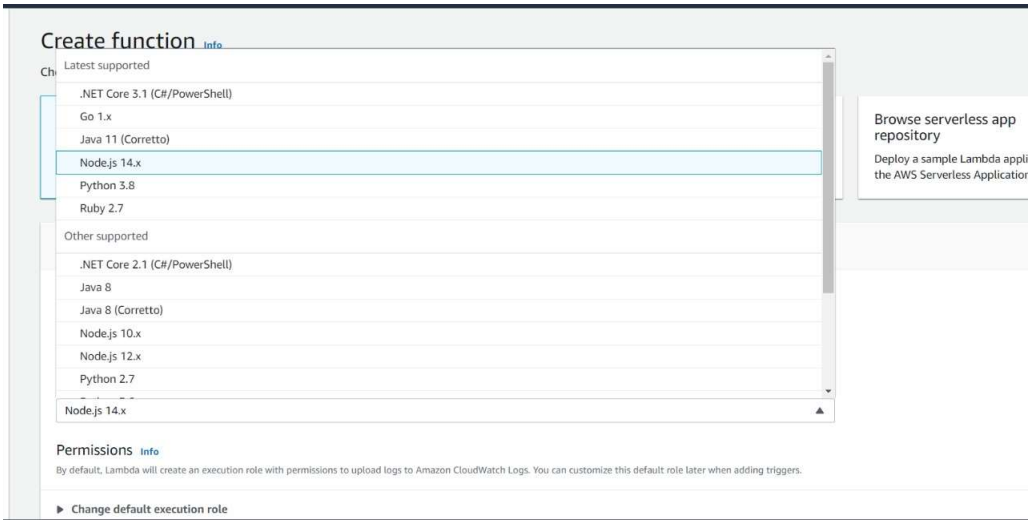
When a client makes a request to one of your API's methods, API Gateway calls your Lam bda authorizer, which takes the caller's identity as input and returns an IAM policy as outp ut.



8. Lambda

AWS Lambda is a serverless compute service that runs your code in response to events an d automatically manages the underlying compute resources for you. You can use AWS La mbda to extend other AWS services with custom logic, or create your own back-end services that operate at AWS scale, performance, and security.

After you upload your code to AWS Lambda, you can associate your function with specifi c AWS resources (e.g. a particular Amazon S3 bucket, Amazon DynamoDB table, Amazo n Kinesis stream, or Amazon SNS notification). Then, when the resource changes, Lambd a will execute your function and manage the compute resources as needed in order to keep up with incoming requests.



9. Dynamo DB

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. DynamoDB lets you offload the administrative burdens of operating and scaling a distributed database so that you don't have to worry about hardware provisioning, setup and configuration, replication, software patching, or cluster scaling. DynamoDB also offers encryption at rest, which eliminates the operational burden and complexity involved in protecting sensitive data. For more information, see DynamoDB Encryption at Rest.

With DynamoDB, you can create database tables that can store and retrieve any amount of data and serve any level of request traffic. You can scale up or scale down your tables' throughput capacity without downtime or performance degradation. You can use the AWS Management Console to monitor resource utilization and performance metrics.

Core Components of Amazon DynamoDB

When you create a table, in addition to the table name, you must specify the primary key of the table. The primary key uniquely identifies each item in the table, so that no two items can have the same key.

DynamoDB supports two different kinds of primary keys:

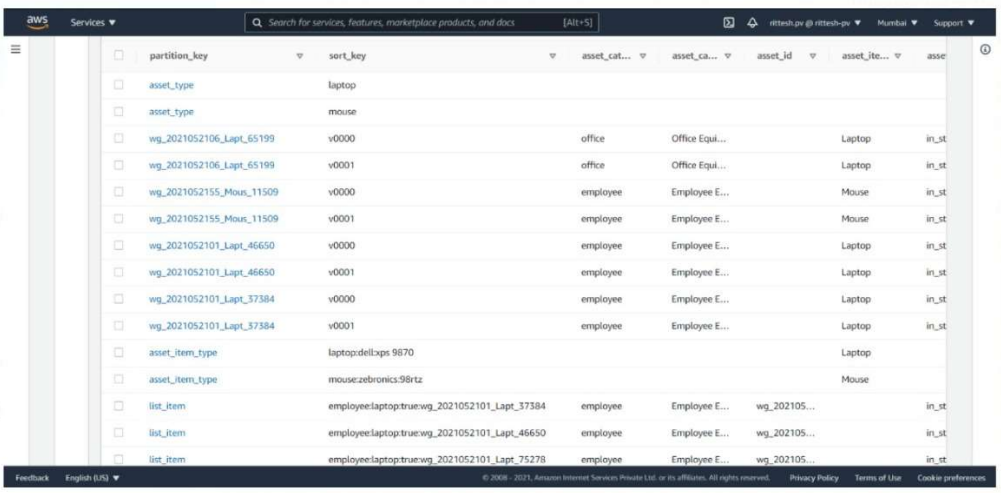
Partition key – A simple primary key, composed of one attribute known as the partition key.

DynamoDB uses the partition key's value as input to an internal hash function. The output from the hash function determines the partition (physical storage internal to DynamoDB) in which the item will be stored.

Partition key and sort key – Referred to as a composite primary key, this type of key is composed of two attributes. The first attribute is the partition key, and the second attribute is the sort key.

DynamoDB uses the partition key value as input to an internal hash function. The output from the hash function determines the partition (physical storage internal to DynamoDB) in which the item will be stored. All items with the same partition key value are stored together, in sorted order by sort key value.

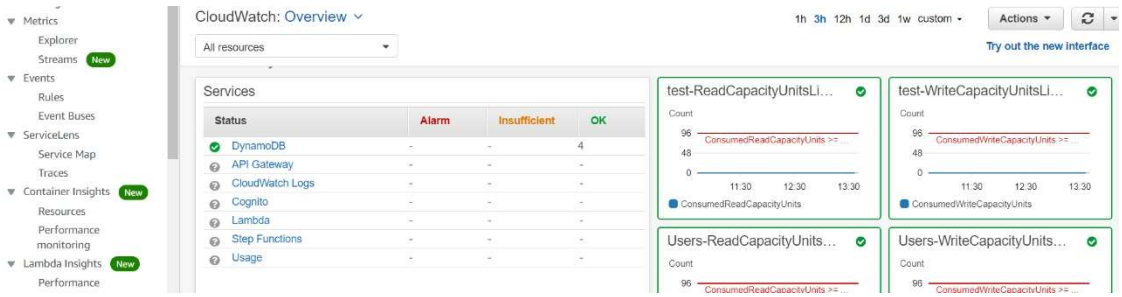
In a table that has a partition key and a sort key, it's possible for two items to have the same partition key value. However, those two items must have different sort key values.



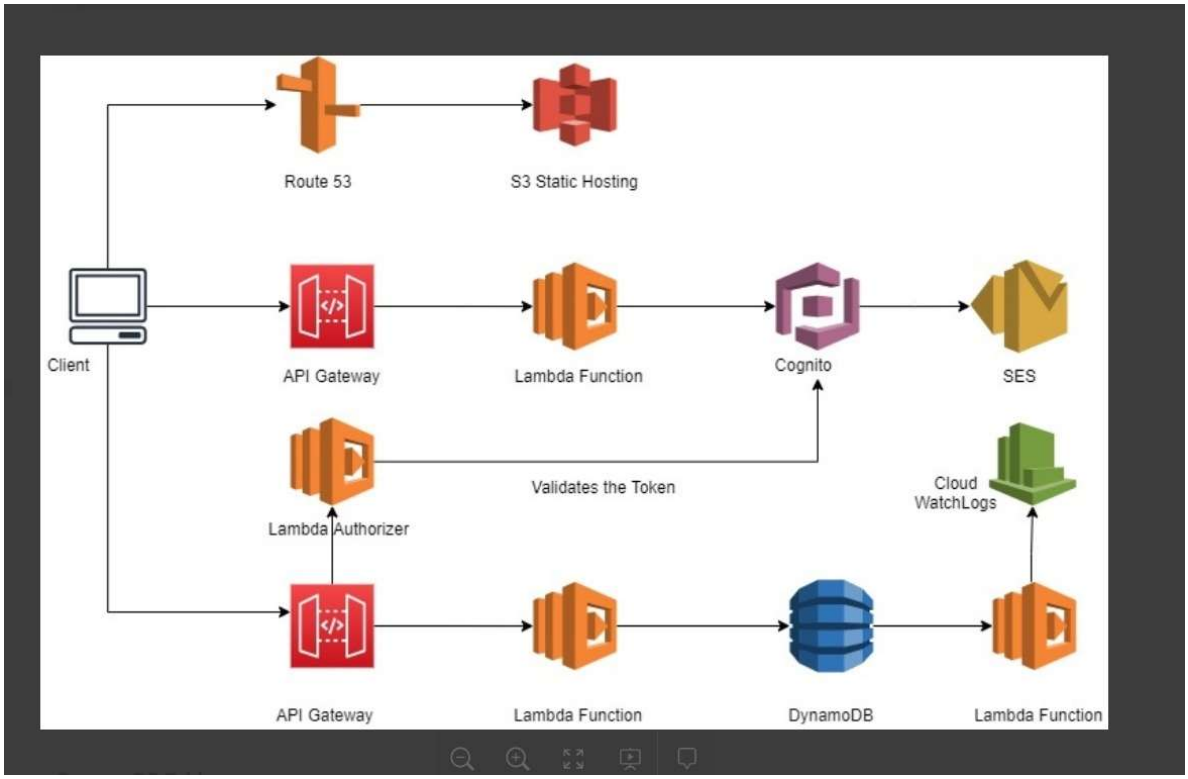
10. CloudWatch Logs

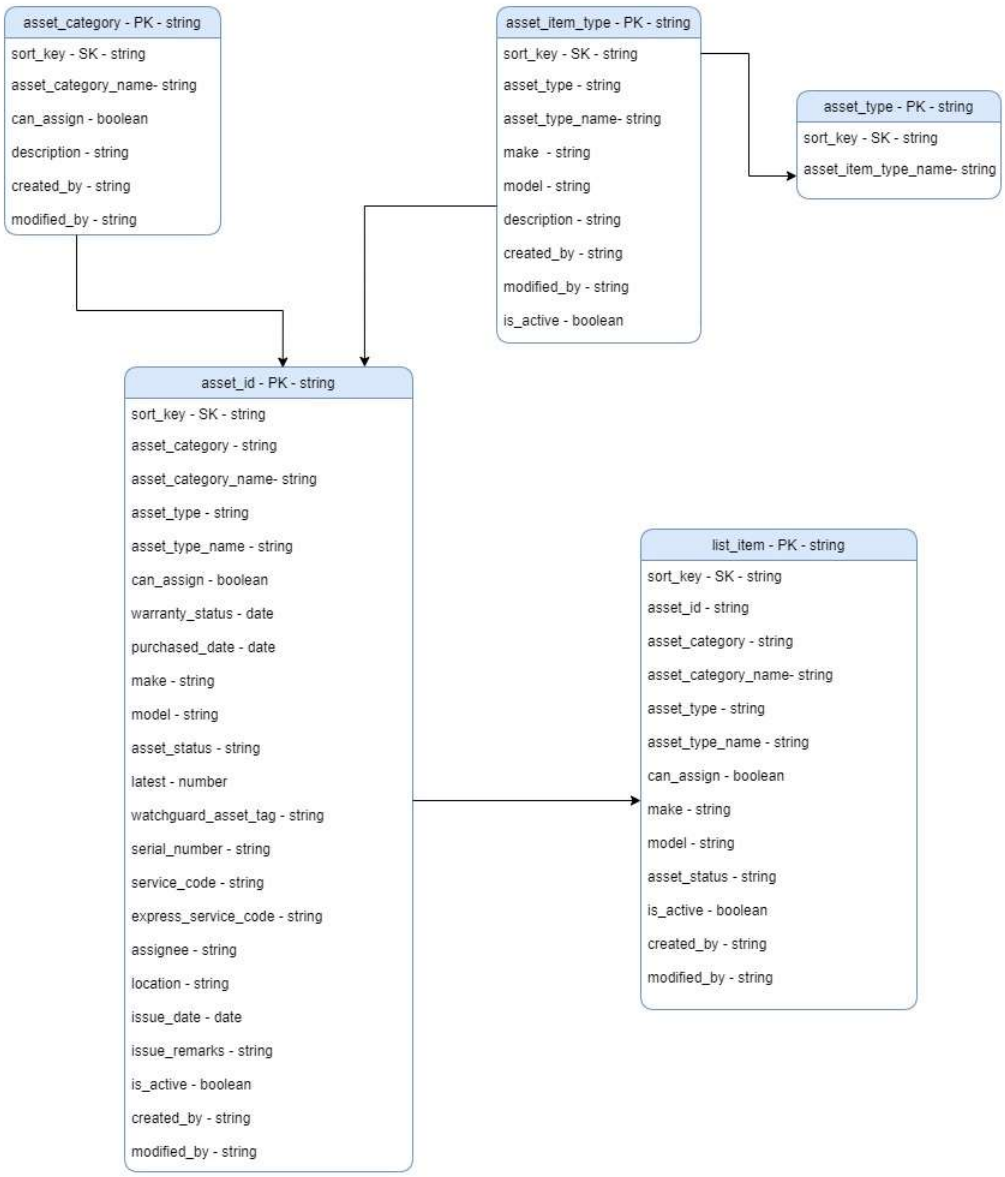
CloudWatch Logs lets you monitor and troubleshoot your systems and applications using your existing system, application and custom log files. With CloudWatch Logs, you can m onitor your logs, in near real time, for specific phrases, values or patterns.

CloudWatch Logs enables you to see all of your logs, regardless of their source, as a singl e and consistent flow of events ordered by time, and you can query them and sort them ba sed on other dimensions, group them by specific fields, create custom computations with a powerful query language, and visualize log data.



- **Architecture Diagram of the Project**




asset_category - PK - string
sort_key - SK - string
asset_category_name- string
can_assign - boolean
description - string
created_by - string
modified_by - string
asset_item_type - PK - string
sort_key - SK - string
asset_type - string
asset_type_name- string
make - string
model - string
description - string
created_by - string
modified_by - string
is_active - boolean
asset_type - PK - string
sort_key - SK - string
asset_item_type_name- string
asset_id - PK - string
sort_key - SK - string
asset_category - string
asset_category_name- string
asset_type - string
asset_type_name - string
can_assign - boolean
warranty_status - date
purchased_date - date
make - string
model - string
asset_status - string
latest - number
watchguard_asset_tag - string
serial_number - string
service_code - string
express_service_code - string
assignee - string
location - string
issue_date - date
issue_remarks - string
is_active - boolean
created_by - string
modified_by - string
list_item - PK - string
sort_key - SK - string
asset_id - string
asset_category - string
asset_category_name- string
asset_type - string
asset_type_name - string
can_assign - boolean
make - string
model - string
asset_status - string
is_active - boolean
created_by - string
modified_by - string

- **API Docs**

1. Asset Category

○ ADD

Endpoint	Method	Sample Request	Sample Response	Success Code	Error Code

/asset-category	POST	{
		"asset_category_name" : "Pantry Equipment",
		"description" : "This Equipment is for the office Pantry use",
		"can_assign" : false,
		"created_by" : "Admin"
		}
			{
			"code": 0,
			"message": "Asset Category added Successfully",
			"added_data": {
			"partition_key": "asset_category",
			"sort_key": "office",
			"asset_category_name": "Office Equipment",
			"description": "This Equipment is for the Office use only",
			"can_assign": false,
			"created_by": "Admin|2021-05-18 09:18",
			"modified_by": "Admin|2021-05-18 09:18"
			}
			}
				1. code : 0
				2. Description : It denotes asset category creation was Successful.
					1. code : 1
					2. Description : It denotes the API Call has Failed due to some Error.

○ FETCH

Fetch Asset Category

Endpoint	Method	Sample Request	Sample Response	Success Code	Error Code
/asset-category	GET	Nil	{ "code": 0, "message": "Asset Categories fetched Successfully", "data": [{ "asset_category_name": "Employee Equipment", "can_assign": true, "partition_key": "asset_category", "created_by": "Admin|2021-05-18 09:19", "sort_key": "employee", "modified_by": "Admin|2021-05-18 09:19", "description": "This Equipment is for the employee use" }, { "asset_category_name": "Office Equipment", "can_assign": false, "partition_key": "asset_category", "created_by": "Admin|2021-05-18 09:18", "sort_key": "office",	1. code : 0 2. Description : It denotes the asset-category fetching was Successful.	1. code : 1 2. Description : It denotes the API Call has Failed due to some Error.

○ UPDATE

Endpoint	Method	Sample Request	Sample Response	Success Code	Error Code

/asset-category	PATCH	{
		"sort_key": "office",
		"data": {
		"description" : "This Equipment is for Office use",
		"modified_by" : "Admin"
		}
		
		}
			{
			"code": 0,
			"message": "Asset Category updated Successfully",
			"updated_data": {
			"asset_category_name": "Employee Equipment",
			"can_assign": true,
			"partition_key": "asset_category",
			"created_by": "Admin|2021-05-18 09:19",
			"sort_key": "employee",
			"modified_by": "Admin|2021-05-18 09:20",
			"description": "This Equipment is for employee use"
			}
			}
				1. code : 0
				2. Description : It denotes the asset category updation was Successful.
					1. code : 1
					2. Description : It denotes the API Call has Failed due to some Error.

2. Asset Item Type

○ ADD

Endpoint	Method	Sample Request	Sample Response	Success Code	Error Code

/asset-item-type	POST	{
		"asset_item_type_name" : "Server",
		"description" : "This is a sample description for this Server",
		"make" : "Cisco",
		"model" : "SG-454",
		"created_by" : "Admin"
		}
			{
			"code": 0,
			"message": "Asset Item Type added Successfully",
			"added_data": {
			"partition_key": "asset_item_type",
			"sort_key": "server:cisco:sg-454",
			"asset_type": "server",
			"asset_type_name": "Server",
			"is_active": true,
			"description": "This is a sample description for this Server",
			"make": "Cisco",
			"model": "SG-454",
			"created_by": "Admin|2021-05-18 09:22",
			"modified_by": "Admin|2021-05-18 09:22"
			}
			}
				1. code : 0
				2. Description : It denotes the asset-item-type creation was Successful.
					1. code : 1
					2. Description : It denotes the API Call has Failed due to some Error.

○ FETCH

Endpoint	Method	Sample Request	Sample Response	Success Code	Error Code
/asset-item-type	GET	Nil	{ "code": 0, "message": "Asset Item types fetched Successfully", "data": [{ "model": "XPS 1970", "partition_key": "asset_item_type", "created_by": "Admin|2021-05-18 09:23", "sort_key": "laptop:dell:xps 1970", "is_active": true, "modified_by": "Admin|2021-05-18 09:23", "asset_type": "laptop", "description": "This is a sample description for this Laptop", "asset_type_name": "Laptop", "make": "Dell" }, { "model": "5415", "partition_key": "asset_item_type", "created_by": "Admin|2021-05-18 09:23",	1. code : 0 2. Description : It denotes the asset-item-types fetching was Successful.	1. code : 1 2. Description : It denotes the API Call has Failed due to some Error.

○ UPDATE

Endpoint	Method	Sample Request	Sample Response	Success Code	Error Code

/asset-item-type	PATCH	{
		"sort_key" : "laptop:dell:xps 1970",
		"data" : {
		"description" : "i5 - 11th Gen , 512 SSD",
		"modified_by" : "Admin"
		}
		}
			{
			"code": 0,
			"message": "Updated Asset Item Type Successfully",
			"updated_data": {
			"model": "XPS 1970",
			"partition_key": "asset_item_type",
			"created_by": "Admin|2021-05-18 09:23",
			"sort_key": "laptop:dell:xps 1970",
			"is_active": true,
			"modified_by": "Admin|2021-05-18 09:26",
			"asset_type": "laptop",
			"asset_type_name": "Laptop",
			"description": "i5 - 11th Gen , 512 SSD",
			"make": "Dell"
			}
			}
				1. code : 0
				2. Description : It denotes the asset-item-type updating was Successful.
					1. code : 1
					2. Description : It denotes the API Call has Failed due to some Error.

- **DELETE**

Endpoint	Method	Sample Request	Sample Response	Success Code	Error Code

/asset-item-type
 	
(or)
 	
/asset-item-type/delete
	DELETE
	
	PATCH
		{
		"sort_key" : "mouse:iball:54t5",
		"modified_by" : "Admin"
		}
			{
			"code": 0,
			"message": "Deleted Asset Item Type Successfully",
			"deleted_data": {
			"model": "54T5",
			"partition_key": "asset_item_type",
			"created_by": "Admin|2021-05-18 09:24",
			"sort_key": "mouse:iball:54t5",
			"is_active": false,
			"modified_by": "Admin|2021-05-18 09:26",
			"asset_type": "mouse",
			"asset_type_name": "Mouse",
			"description": "This is a sample description for iBall Mouse",
			"make": "iBall"
			}
			}
				1. code : 0
				2. Description : It denotes the asset-item-type deleting was Successful.
					1. code : 1
					2. Description : It denotes the API Call has Failed due to some Error.

3. Asset Item

- **ADD**

Endpoint	Method	Sample Request	Sample Response	Success Code	Error Code

/asset-item	POST	{
		"asset_type": "monitor",
		"asset_category": "employee",
		"quantity": 1,
		"make": "LG",
		"model": "G-Curve",
		"asset_item_type_name" : "Monitor",
		"asset_category_name" : "Employee Equipment",
		"can_assign": true,
		"asset_status": "in_stock",
		"created_by": "Admin",
		"purchased_date": "3/20/2021",
		"warranty_status": "3/20/2022",
		"watchguard_asset_tag" : "WG102",
		"serial_number" : "4654654",
		"service_code" : "SR74865",
		"assignee": "Luke Dimmer",
		"issue_date": "5/12/2021",
		"issue_remark": "Issued for Malware Analysis Projects"
		}
			{
			"code": 0,
			"message": "Added Asset Item Successfully",
			"added_data": [
			{
			"partition_key": "list_item",
			"sort_key": "employee:monitor:true:wg_2021051829_Moni_60554",
			"asset_id": "wg_2021051829_Moni_60554",
			"asset_type": "monitor",
			"asset_category": "employee",
			"asset_type_name": "Monitor",
			"asset_category_name": "Employee Equipment",
			"make": "LG",
			"model": "G-Curve",
			"asset_status": "in_stock",
			"can_assign": true,
			"is_active": true,
			"created_by": "Admin|2021-05-18 09:29",
			"modified_by": "Admin|2021-05-18 09:29"
			}
			]
			}
				1. code : 0
				2. Description : It denotes updation was Successful.
					1. code : 1
					2. Description : It denotes the API Call has Failed due to some Error.

- **FETCH**

Endpoint	Method	Sample Request	Sample Response	Success Code	Error Code

/asset-item	GET	Params: {
		mode=2,
		asset_id=wg_2021051829_Moni_60554
		}
			{
			"code": 0,
			"message": "Asset Items fetched Successfully",
			"data": [
			{
			"model": "G-Curve",
			"asset_category_name": "Employee Equipment",
			"can_assign": true,
			"partition_key": "wg_2021051829_Moni_60554",
			"warranty_status": "3/20/2022",
			"created_by": "Admin|2021-05-18 09:29",
			"assignee": "Luke Dimmer",
			"sort_key": "v0000",
			"service_code": "SR74865",
			"purchased_date": "3/20/2021",
			"modified_by": "Admin|2021-05-18 09:29",
			"make": "LG",
			"serial_number": "4654654",
			"asset_status": "in_stock",
			"issue_date": "5/12/2021",
			"asset_item_type_name": "Monitor",
			"watchguard_asset_tag": "WG102",
				1. code : 0
				2. Description : It denotes items fetched Successful.
					1. code : 1
					2. Description : It denotes the API Call has Failed due to some Error.

- **UPDATE**

Endpoint	Method	Sample Request	Sample Response	Success Code	Error Code

/asset-item	PUT	{
		"asset_id":
		"wg_2021051834_Serv_35732",
		"data": {
		"asset_status": "in_use",
		"watchguard_asset_tag" : "WG101",
		"serial_number" : "68548745",
		"service_tag" : "SR74865",
		"location": "Noida Office, Cabin 238",
		"issue_date": "5/12/2021",
		"issue_remark": "Issued for Server test",
		"modified_by":"Admin"
		}
		}
			{
			"code": 0,
			"message": "Updated Asset Item Successfully",
			"updated_data": {
			"model": "SG-454",
			"asset_category_name": "Office Equipment",
			"can_assign": false,
			"partition_key": "list_item",
			"created_by": "Admin|2021-05-18 09:34",
			"sort_key":
			"office:server:true:wg_2021051834_Serv_35732",
			"modified_by": "Admin|2021-05-18 09:39",
			"asset_id": "wg_2021051834_Serv_35732",
			"make": "Cisco",
			"asset_status": "in_use",
			"asset_category": "office",
			"is_active": true,
			"asset_type": "server",
			"asset_type_name": "Server"
			}
				1. code : 0
				2. Description : It denotes updation was Successful.
					1. code : 1
					2. Description : It denotes the API Call has Failed due to some Error.

- **DELETE**

Endpoint	Method	Sample Request	Sample Response	Success Code	Error Code

/asset-item
 	
(or)
 	
/asset-item/delete
	DELETE
	
	POST
		{
		"asset_id" :
		"wg_2021051834_Serv_61335",
		"modified_by":"Admin"
		}
			{
			"code": 0,
			"message": "Deleted Asset Item Successfully",
			"deleted_data": {
			"partition_key": "list_item",
			"sort_key":
			"office:server:false:wg_2021051834_Serv_61335",
			"asset_id": "wg_2021051834_Serv_61335",
			"asset_type": "server",
			"asset_category": "office",
			"asset_type_name": "Server",
			"asset_category_name": "Office Equipment",
			"make": "Cisco",
			"model": "SG-454",
			"asset_status": "in_stock",
			"can_assign": false,
			"created_by": "Admin|2021-05-18 09:34",
			"modified_by": "Admin|2021-05-18 09:34",
			"is_active": false
			}
			}
				1. code : 0
				2. Description : It denotes the delete was Successful.
					1. code : 1
					2. Description : It denotes the API Call has Failed due to some Error.

4. User

○ SIGN-UP

Sign Up User

Endpoint	Method	Sample Request	Sample Response	Success Codes	Error Codes

/sign-up	POST	{
		"email" : "test@gmail.com"
		}
			{
			"code": 0,
			"message": "Created User successfully"
			}
				1. code : 0
				2. Description : It denotes the user creation was Successful
					1. code : 1
					2. Description : It denotes the API Call has Failed due to some Error

○ SET-PASSWORD

Endpoint	Method	Sample Request	Sample Response	Success Codes	Error Codes

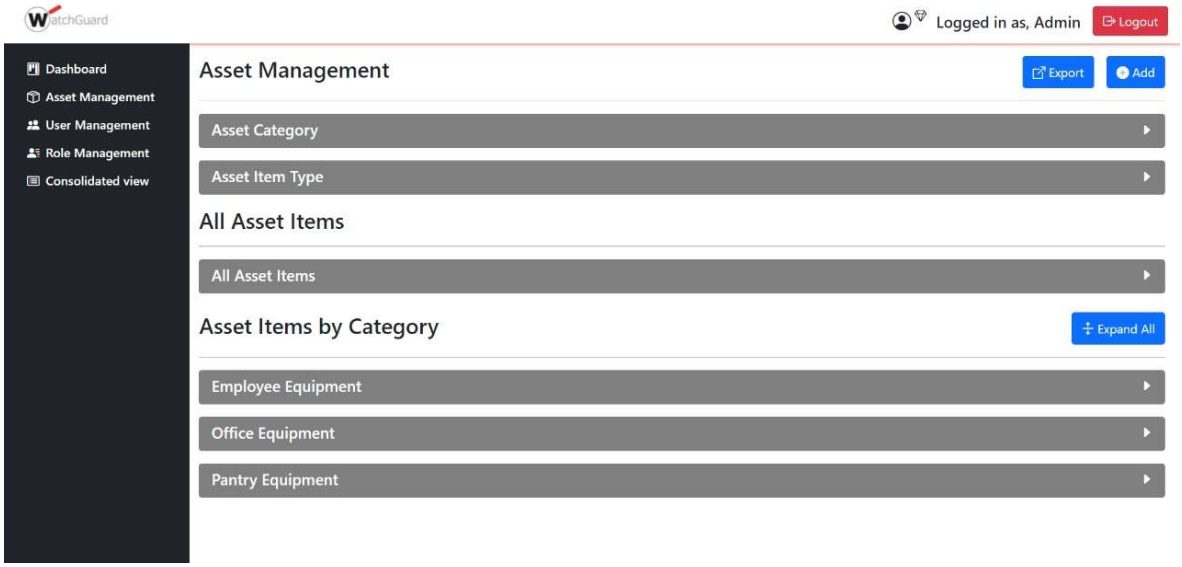
/password	POST	{
		"email" : "test@gmail.com",
		"new_password" : "test"
		}
			{
			"code": 0,
			"message": "User password changed successfully"
			}
				1. code : 0
				2. Description : It denotes that the change in password was successful
					1. code : 1
					2. Description : It denotes the API Call has Failed due to some Error

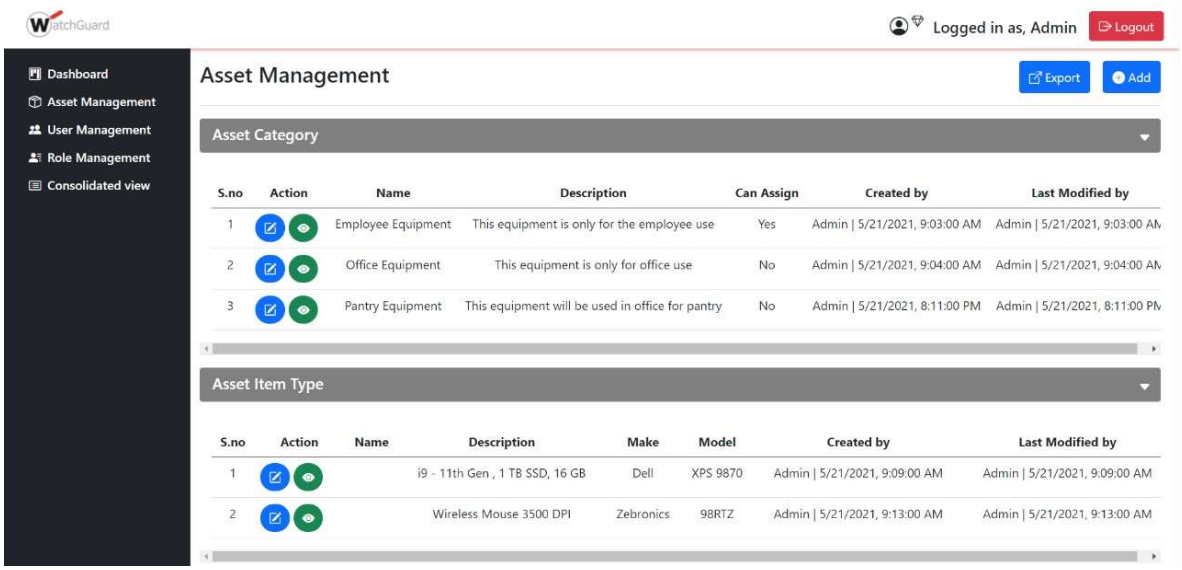
○ SIGN-IN

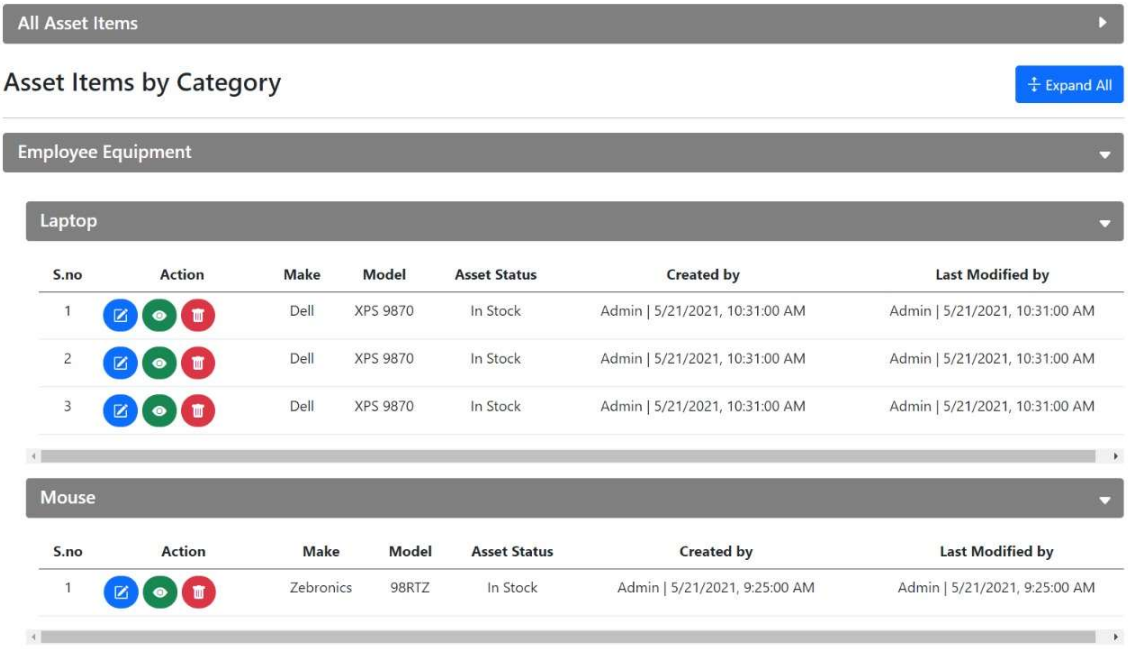
Endpoint	Method	Sample Request	Sample Response	Success Codes	Error Codes

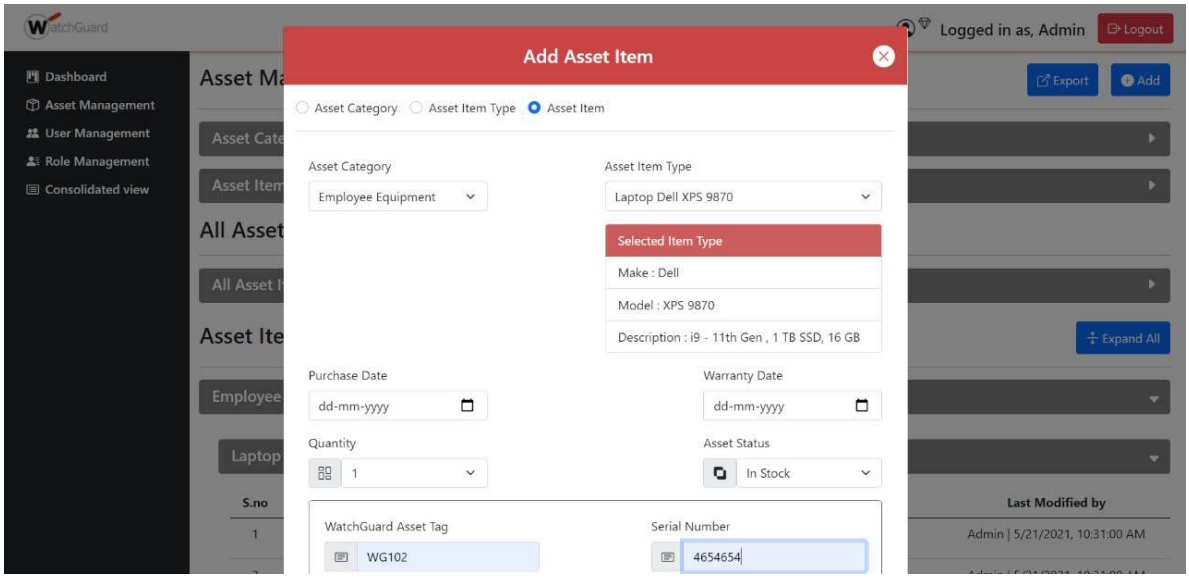
1. /sign-in	POST	{
		"email" : "test@gmail.com",
		"new_password" : "testtest"
		}
			{
			"code": 0,
			"message": "login Success",
			"data": {
			{
			"ChallengeParameters": {},
			"AuthenticationResult": {
			"AccessToken": "eyJraWQiOiJBY1NDUU5YT2xrSDFzUldkWER0a3k5UzlOdXpOYWFtQWdXVGVMU1Z4QVU0PSIsImFsZyI6IlJTMjU2In0[TRUNCATED],
			",
			"ExpiresIn": 3600,
			"TokenType": "Bearer",
			"RefreshToken": "eyJjdHkiOiJKV1QiLCJlbmMiOiJBMjU2R0NNIiwiYWxnIjoiUlNBLU9BRVAifQ[TRUNCATED]",
			"IdToken": "eyJraWQiOiJNRytra2FvbXBKKzhRZWVhTUxDTlFPZXRJWGRTUDdlaEcxZEg2U1wvdE5Xdz0iLCJhbGciOiJSUzI1NiJ9[TRUNCATED]"
				1. code : 0
				2. Description : It denotes that the login was Successful
					1. code : 1
					2. Description : It denotes the API Call has Failed due to some Error

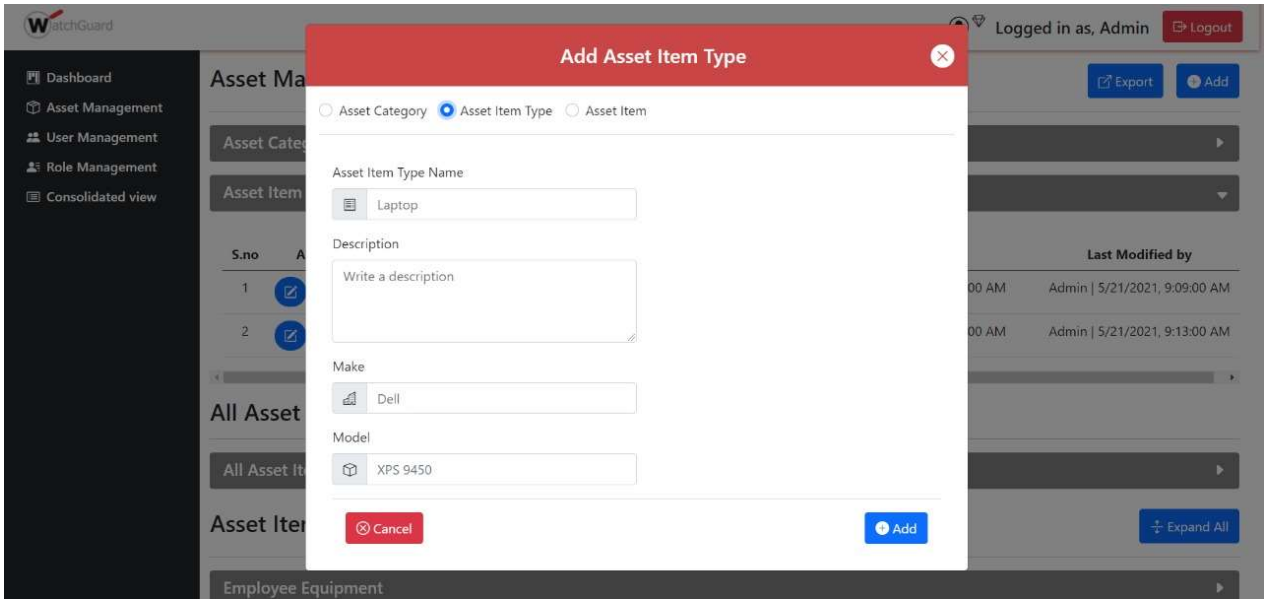
- **UI Designs**

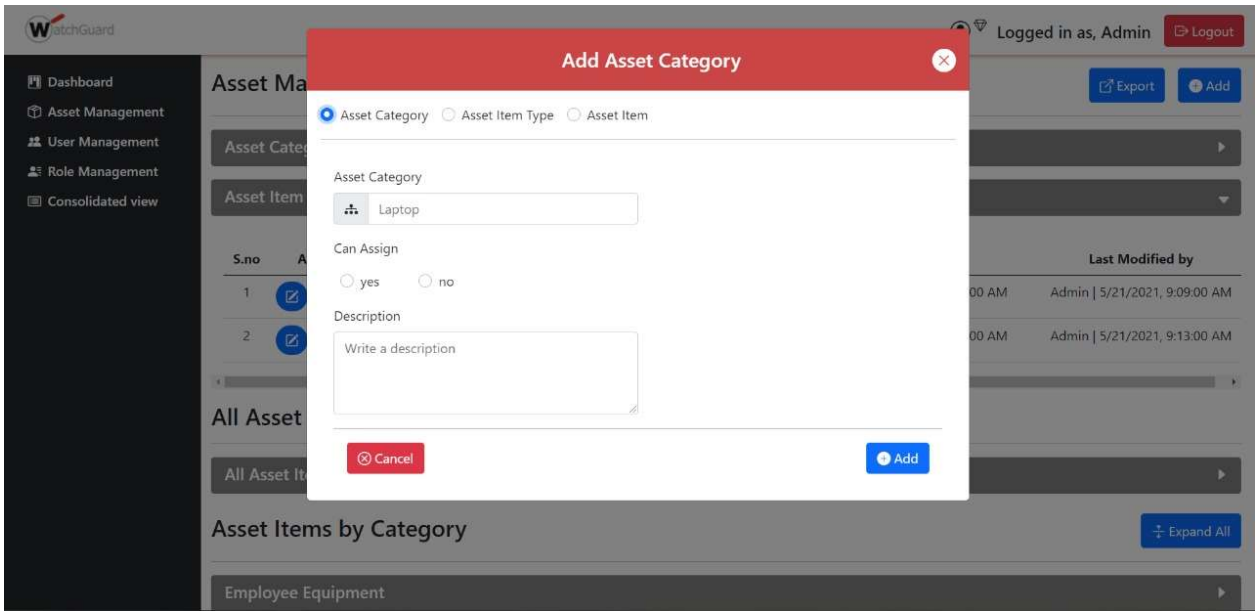

WatchGuard
Logged in as, Admin
Logout
Dashboard
Asset Management
User Management
Role Management
Consolidated view
Asset Management
Export
Add
Asset Category
Asset Item Type
All Asset Items
All Asset Items
Asset Items by Category
Expand All
Employee Equipment
Office Equipment
Pantry Equipment


WatchGuard
Logged in as, Admin
Logout
Dashboard
Asset Management
User Management
Role Management
Consolidated view
Asset Management
Export
Add
Asset Category
S.no Action Name Description Can Assign Created by Last Modified by
1 Employee Equipment This equipment is only for the employee use Yes Admin | 5/21/2021, 9:03:00 AM Admin | 5/21/2021, 9:03:00 AM
2 Office Equipment This equipment is only for office use No Admin | 5/21/2021, 9:04:00 AM Admin | 5/21/2021, 9:04:00 AM
3 Pantry Equipment This equipment will be used in office for pantry No Admin | 5/21/2021, 8:11:00 PM Admin | 5/21/2021, 8:11:00 PM
Asset Item Type
S.no Action Name Description Make Model Created by Last Modified by
1 i9 - 11th Gen , 1 TB SSD, 16 GB Dell XPS 9870 Admin | 5/21/2021, 9:09:00 AM Admin | 5/21/2021, 9:09:00 AM
2 Wireless Mouse 3500 DPI Zebronics 98RTZ Admin | 5/21/2021, 9:13:00 AM Admin | 5/21/2021, 9:13:00 AM


All Asset Items
Asset Items by Category
Expand All
Employee Equipment
Laptop
S.no Action Make Model Asset Status Created by Last Modified by
1 Dell XPS 9870 In Stock Admin | 5/21/2021, 10:31:00 AM Admin | 5/21/2021, 10:31:00 AM
2 Dell XPS 9870 In Stock Admin | 5/21/2021, 10:31:00 AM Admin | 5/21/2021, 10:31:00 AM
3 Dell XPS 9870 In Stock Admin | 5/21/2021, 10:31:00 AM Admin | 5/21/2021, 10:31:00 AM
Mouse
S.no Action Make Model Asset Status Created by Last Modified by
1 Zebronics 98RTZ In Stock Admin | 5/21/2021, 9:25:00 AM Admin | 5/21/2021, 9:25:00 AM


Logged in as, Admin
Logout
Dashboard
Asset Management
User Management
Role Management
Consolidated view
Export
Add
Add Asset Item
Asset Category
Asset Item Type
Asset Item
Asset Category
Employee Equipment
Asset Item Type
Laptop Dell XPS 9870
Selected Item Type
Make : Dell
Model : XPS 9870
Description : i9 - 11th Gen , 1 TB SSD, 16 GB
Purchase Date
dd-mm-yyyy
Warranty Date
dd-mm-yyyy
Quantity
1
Asset Status
In Stock
WatchGuard Asset Tag
WG102
Serial Number
4654654
Expand All
Last Modified by
Admin | 5/21/2021, 10:31:00 AM


Add Asset Item Type
Asset Category
Asset Item Type
Asset Item
Asset Item Type Name
Laptop
Description
Write a description
Make
Dell
Model
XPS 9450
Cancel
Add
Logged in as, Admin
Logout
Dashboard
Asset Management
User Management
Role Management
Consolidated view
Export
Add
Expand All
Employee Equipment
Last Modified by
Admin | 5/21/2021, 9:09:00 AM
Admin | 5/21/2021, 9:13:00 AM


Add Asset Category
Asset Category
Asset Item Type
Asset Item
Asset Category
Laptop
Can Assign
yes
no
Description
Write a description
Cancel
Add
Logged in as, Admin
Logout
Dashboard
Asset Management
User Management
Role Management
Consolidated view
Export
Add
Expand All
Asset Items by Category
Employee Equipment
Last Modified by
Admin | 5/21/2021, 9:09:00 AM
Admin | 5/21/2021, 9:13:00 AM

Chapter - 4

RESULTS AND CONCLUSION

This internship gave me a clear picture of how working in real world looks like, it has increased my knowledge to a great extent. In addition to the technical knowledge I gained, I also learned what are the problems we face in day-to-day life when we work on any project and how we manage to solve those issues, what benefits we have while working in a team and how to balance between the different responsibilities we have.

By making this report on AWS and its services, I got to know about the need of this technology in today's world.

It is very important to manage such a huge amount of data that is prevailing around us and technologies such as AWS are the ones that are exceeding in doing so.

They remove the problems of big data and also tell the efficient ways to store, compute and analyze your client's data.

I also got to know about the various new terms and that one can have a great future and scope in this technology.

Working on AWS and gaining the hands-on experience on its working has been a very enriching journey for me.

References

[1] https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html

[2] https://ap-south-1.console.aws.amazon.com/lambda/home?region=ap-south-1#/functions'

[3] https://ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#Instances:instanceState=running

[4] https://ap-south-1.console.aws.amazon.com/cloudwatch/home?region=ap-south-1

[5] https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html'

[6] https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-identity-pools.html

[7] https://ap-south-1.console.aws.amazon.com/apigateway/main/apis?region=ap-south-1

[8] https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/kms.html

[9] https://medium.com/petabytz/amazon-web-services-wordpress-configuration-on-ecs-using-ecr-images-9c1e97d05aaa