



UNIVERSITY OF SRI JAYEWARDENEPURA
Faculty of Engineering
CO3600 Industrial Training

TRAINING REPORT

Training Organization

Mitra Innovation (Pvt.) Ltd.

Name	:	U.D.P.D Uduwela
Student ID	:	18/ENG/112
Department	:	Computer Engineering
Training Period	:	From 02/02/2022 to 05/08/2022

Acknowledgments

First, I want to thank Dr. Thilakasiri Bandara (Dean Faculty of the Engineering University of Sri Jayewardenepura), Dr. Randima Dinalankara (Head of the Department of Computer Engineering, University of Sri Jayewardenepura), and Dr. Udaya Wijenayake (Industrial Training Coordinator of the Department), all the lectures of the department of computer Engineering and all the others who not mentioned here for their support in getting the internship opportunity at Mitra Innovation.

On the other hand, I want to thank the Industrial Training Division of the Faculty of Engineering, the University of Sri Jayewardenepura, and the National Apprentice and Industrial Training Authority for organizing the internship programs and their supervision during the Internship.

The industrial training experience I got from Mitra Innovation was a tremendous Professional experience in my life. I had the chance to learn new frameworks and apply what I learned in the faculty. On the other hand, I had the chance to work on a client project at Mitra innovation. Therefore I would Like to express my sincere gratitude to Dr. Ashok Suppiah (Group C.E.O.), Mr. Thilina Herath (Principal Architect), who created lots of opportunities and guided us through the First project, Mr. Dileema Fernando (Senior Tech lead) who help me to identify my weakness and help me to correct those within a shorter period, Mr. Gayan Madhusanka(Senior Software Engineer), who support me and encourage me in work at Trion project, Mr. Shimantha Perera(Project Manager) who played the scrum master role in the first project, Mrs. Saduni Maduhansani(Associate Software Engineer) who help me a lot throughout my Internship, Mr. Thilina kularathna(Software architect), who was the key person to guide us in the Second project and all the other team members at Dygisec who supported me to complete my industrial training successfully.

Finally, I would like to thank Mrs. Chathurika Gunawardena (Head of Talent Acquisition), and the Human Resource Team, especially those who support us and check our status at Mitra Innovation

Preface

This report mainly describes my industrial training program at Mitra Innovation (Pvt.) Ltd., Moratuwa. I worked for 27 weeks, from 02/02/2022 to 05/08/2022. Though we acquired lots of engineering knowledge in the computer engineering discipline over the three years of our academic work, due to the covid 19 circumstances, we were not able to do practicals physically. Therefore, some important concepts were missed during the online practical using simulators and so on. In the Industry, we mainly need practical knowledge. Therefore, an industrial training program is an excellent chance to identify your weaknesses and correct them before going to the actual Industry. Through this training program, you can improve employability skills such as applying conceptual knowledge to the workplace, professional working practices, employee interactions, and professional ethics. This report consists of three main chapters. The first chapter illustrates the company overview, including history, vision, and mission. The second chapter is about the technical and non-technical things I have experienced at the company during my industrial Training program. The last chapter is about the conclusion of the entire report. Specially I have added some valued comments for the last chapter to make this program a meaningful one for future batches at the faculty of the Engineering University of Sri Jayewardenepura.

Table of Contents

List of Figures.....	6
List of Tables	8
List of Acronyms	9
Chapter 1 -Introduction	10
1.1. About Training Organization	10
1.1.1. Nature Of Mitra Innovation.....	10
1.1.2. Vision and mission at Mitra innovation.....	11
1.1.3. Current Company Partnerships	11
1.1. Brief View of Organizational Chart.....	11
1.1.1. Hierarchical Levels	12
1.1.2. Context of Mitra Innovation	13
1.2. Recruitment Procedures.....	14
1.3. Absence / Leave	14
1.3.1. Annual Leave.....	14
1.3.2. Other Leave	14
1.4. Pay and Benefits.....	15
1.5. SWOT Analysis	15
1.5.1. Strengths	15
1.5.2. Weaknesses	15
1.5.3. Opportunities.....	16
1.5.4. Threats	16
Chapter 2 - Training Experience.....	16
2.1. Short description of my first project (Triton)	16
2.1.1. Introduction.....	16
2.1.2. Stakeholders	17
2.1.3. Tools and techniques.....	17
2.1.4. Knowledge regarding GIT	18
2.1.4.1. Basic GIT operations used in the project.....	18
2.2. Tasks Assigned in the First project	18
2.2.1. Task 1- Golang REST API development for compliance dashboard.....	18
2.2.2. Task 2 – Swagger API documentation for Services	23
2.2.3. Task 3 -Done research on creating an O.I.D.C. plugin for the Kong gateway	27
2.2.4. Task 4 – Update the Policy Compliances-sample Data Feed v1.00.xlsx	30
2.2.5. Task 5 – Make a Golang program to read CSV files and extract data into struts	31
2.2.6. Task 6 – Modified the current E.R.D. of compliance rules repository.....	32

2.2.7.	Task 7–Update Golang data seeder according to the new changes	34
2.2.8.	Task 8 – Fill out the A.W.S. managed config rules repository.....	38
2.2.9.	Task 9 – Make an ETL to update data from the tsv format file to Cloud Database	40
2.2.10.	Task 10 – Make a YAML to automate the ETL deploying process	46
2.3.	Short overview of the Second Project (ATS -Application Tracking System).....	48
2.4.	Tasks Assigned in the First project	49
2.4.1.	Task 1 – Make an ER diagram and a scheduler mechanism for the database.....	49
2.4.2.	Task 2 – propose some wireframes to the team.....	53
2.4.3.	Task 3 – Supported creating a 3P estimation sheet	54
2.4.4.	Task 4 – Involved in creating the Project Timeline and Gantt chart.....	55
<u>2.5.</u>	<u>Non-Technical Experience Gained at Mitra Innovation</u>	<u>56</u>
Chapter 3 -Conclusion.....		56
3.1.	Internship Experience at Mitra Innovation.....	56
3.2.	Self-Assessment on the Internship.....	56
3.3.	Suggestions.....	57
List of References.....		57

List of Figures

- Figure 1: Mitra Innovation Logo
- Figure 2: View of the organizational chart
- Figure 3: Mitra Innovation Hierarchy
- Figure 4: Context of Mitra Innovation
- Figure 5: Recruitment process stages
- Figure 6: Check the environment variables set in WSL
- Figure 7: After installing the necessary extensions in VSCode
- Figure 8: Successful build of the frontend repository
- Figure 9: Microservice Backend component design Structure
- Figure 10: The Successful build of the back-end (Pod & services details)
- Figure 11: The successful build of the back-end (replica set & deployments details)
- Figure 12: Testing the API endpoints using Postman IDE
- Figure 13: Due to poor Internet connection, Some docker files failed to build
- Figure 14: packages imported into handler.go file
- Figure 15: General API Annotations in main.go file
- Figure 16: API annotation in handler.go file
- Figure 17: Final Successful swag init command
- Figure 18: Initial Loading Overview of SwaggerUI
- Figure 19: PUT method API Endpoint
- Figure 20: GET method API Endpoint
- Figure 21: Programmatically set the swagger details
- Figure 22: Proposed Workflow of the plugin
- Figure 23: The Overview of the Konga dashboard
- Figure 24: After Doing Modification On the Kong Gateway, the gateway crashed every time.
- Figure 25: 11 tables and their names:
- Figure 26: Seeder helper workflow
- Figure 27: The generated sdata.go file
- Figure 28: Modified and optimized ER diagram
- Figure 29: dao.go file where the database Types were written
- Figure 30: After several modifications seeder successfully workedFigure 32:
- Figure 31: Connected to the Posgredql database image using K8s port forward method

Figure 32: Policy_rules table and its relations

Figure 33: risk_categories table and its relations

Figure 34: Error occurred due to the unavailability of some libraries installed in the working environment.

Figure 35: Comment on some lines in the service.mk file to escape the error temporally

Figure 36: Error occurred due to Incorrect table relations

Figure 37: Error occurred due to insufficient variable size

Figure 38: Conditions and its relevant command for S3-bucket read prohibited

Figure 39: Truth tabel design for the relavant rule

Figure 40: Simple diagram that shows the working flow of the ETL

Figure 41: Workflow diagram On Creating IAM Custom Role

Figure 42: Flow diagram to Get the event JSON response using CloudwatchLogs(Not necessary we can obtain those data using the AWS dashboard)

Figure 43: Triton ETL WorkFlow Diagram

Figure 44: Online IDE, which AWS provides us to test our lambda functions

Figure 45: Time Out Error occurred while developing the ETL

Figure 46: Due to commas inside paragraphs, some characters were missing after loading the data to the database.

Figure 47: YAML file of the ETL

Figure 48: After connecting the database todbeaver

Figure 49: AWS Monthly bill Summary

Figure 50: AWS Services used and their charges

Figure 51: The ER diagram I designed for the Applicant tracking project

Figure 52: The part of data captured in the pipeline stages table

Figure 53: Workflow diagram of the scheduler

Figure 54: ATS Deployment Diagram

Figure 55: APIS created from the PeopleHR end to get data to our database.

Figure 56 - 61: ATS Wireframes

Figure 62: Mitra 3P estimation Sheet

Figure 63: ATS Grant chart(due to the space limitation only includes the first 23 tasks)

List of Tables

Table 1: Current Partnership of the company

Table 2: Stakeholders details of the Triton project

Table 3: Tools and techniques Used in project

Table 4: Git commands Used in the Project

Table 5: Details On Kong Gateway, Keycloak, and Microservice architecture

Table 6: Policy & compliance table details

List of Acronyms

CI/CD	: Continuous Integration and Continuous Deployment
A.W.S.	: Amazon Web Services
SDLC	: Software Development Life Cycle
B.A.	: Business Analyst
PM	: Project Manager
YAML	: Yet Another Markup Language
REST	: Representational State Transfer
API	: Application Programming Interface
JSON	: JavaScript Object Notation
A.W.S.	: Amazon Web Services
W.S.L.	: Widows Subsystem For LINUX
EPF	: Employee Provident Fund
ETF	: Employee Trust Fund
C.O.O.	: Chief Operating Officer
C.E.O.	: Chief Executive Officer
MD	: Managing Director
QA	: Quality Assurance
B.A.	: Business Analysis
PMO	: Project Management Officer
C.D.O.	: Chief Data Officer
C.F.O.	: Chief Financial Officer
C.P.O.	: Chief Product Officer
ATS.	: Applicant Tracking System
RDS.	: Relational Database Service
ETL.	: Extract Transform Load
TSV	: Tab Separated Values
N.A.I.T.A.	: National Apprentice and Industrial Training Authority

Chapter 1 -Introduction

1.1. About Training Organization

Mita Innovation (Pvt) Limited is a privately-owned company registered in Sri Lanka. It operates as an independent company with its shareholders and governance structures. The company began operations with a skeleton staff, one office, and two clients and has since grown to a global organization with 350+ employees and six offices - three in Sri Lanka, a Headquarters in London, an office in the United States, an Australian presence, and a new office in Suffolk, England. All engineering research and development happens in Colombo, close to rawatawatta university premises.

The company comprises specialists in product incubation, digital transformation, and Cloud-to-Cloud Systems integration, with expert skills in WSO2 and A.W.S. technologies. The company plans to double its engineering team during the latter part of this year.



Figure 1-Mitra Innovation Company Logo

1.1.1. Nature Of Mitra Innovation

Mitra innovation is a technology innovation company based in Sri Lanka, Australia, the U.S., and the U.K. in 2012 by three co-founders. They are-

1. Dr. Ashok Suppiah (Group C.E.O. & Co-founder Mitra Group)
2. Derek Bell (Founder and C.O.O. at Mitra Innovation)
3. Dammika Ganegama (Co-Founder & Executive Vice President – Mitra Group)

Mitra Innovation has two central business Units. They are-

- 1. Mitra Digital** -Focuses on software service delivery
- 2. Mitra ventures** – Focuses on product innovation and partnerships with other joint ventures

1.1.2. Vision and mission at Mitra innovation

In the Mitra innovation, the founders have only one single vision. Which is to help entrepreneurs and enterprises build and launch disruptive technology solutions. Mitra Innovation's Mission was to help as many successful products and ventures as possible.

1.1.3. Current Company Partnerships

Table 1-Current Partnership of the company

Partner Organization	Type of partnership	Area
WSO2	Preferred / Certified Partner	Integration
Amazon Web Services	Standard Consulting Partner	Cloud infrastructure
Intrexx	Partner	Data Integration
Google Cloud Platform	Potential partnership	Cloud platform
Talend	Potential partnership	Data Integration
NBC CAPITAL	partner	Australian business generation
Beyond Analysis	partner	Business Analytics
Jacob Bailey	partner	Design Agency

1.1. Brief View of Organizational Chart

Here this was the part of the organizational chart, and from the green colour, some of our team members who worked for the Triton project were represented, and from the red colour, I have pointed out where I am in the chart. All the 400+ employees have a specific place inside the chart, as shown below. Due to the space limitation, I have only drawn a tiny part of the organizational chart's tree

structure shown in the PeopleHR system. In Mitra, I was working under Mitra digital services.

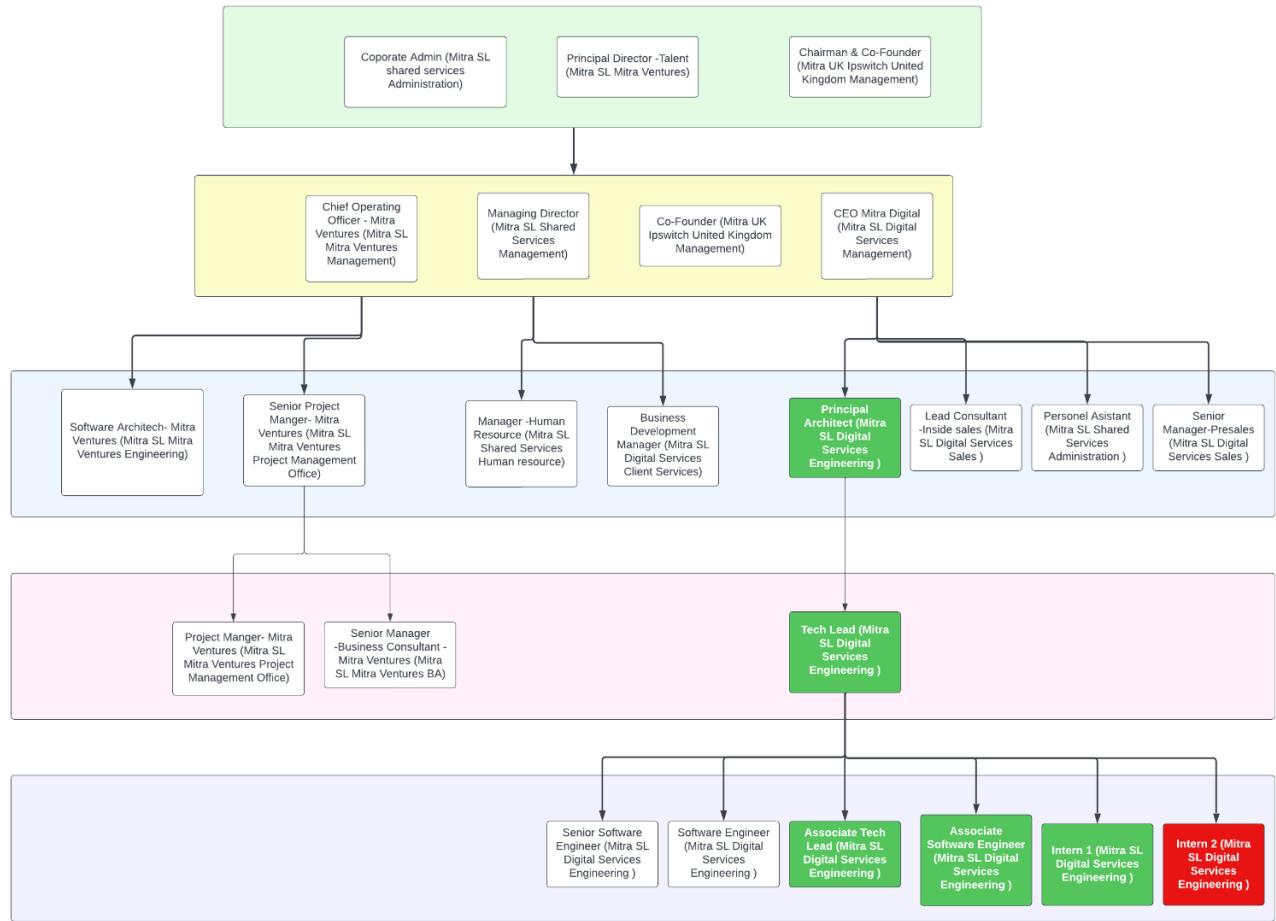


Figure 2- View of the organizational chart

1.1.1. Hierarchical Levels

Mitra Innovations Only follows a hierarchy in administration, accounting, and HR works but follows a very flat hierarchy in the software development life cycle. In administration, all Mitra Innovation employees have people leaders to whom they report. But in development, all the team members are treated as a single unit and headed by a scrum master, a few software Architects, and senior Tech Leads. Above them, there are one or two Senior Software Architects who overlook a single project.

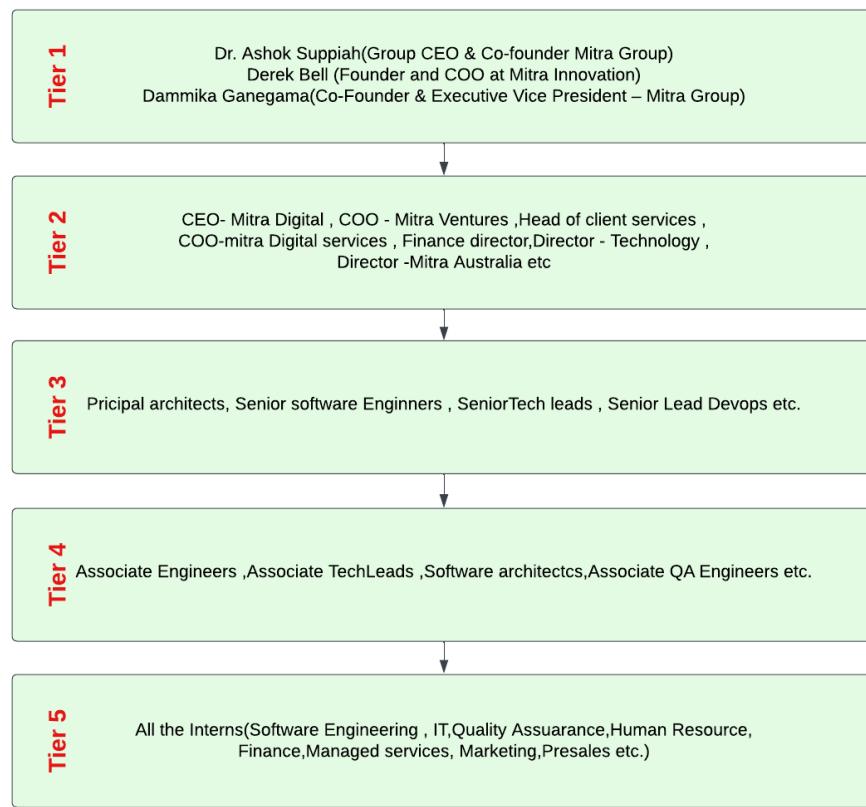


Figure 3-Mitra Innovation Hierarchy

1.1.2. Context of Mitra Innovation

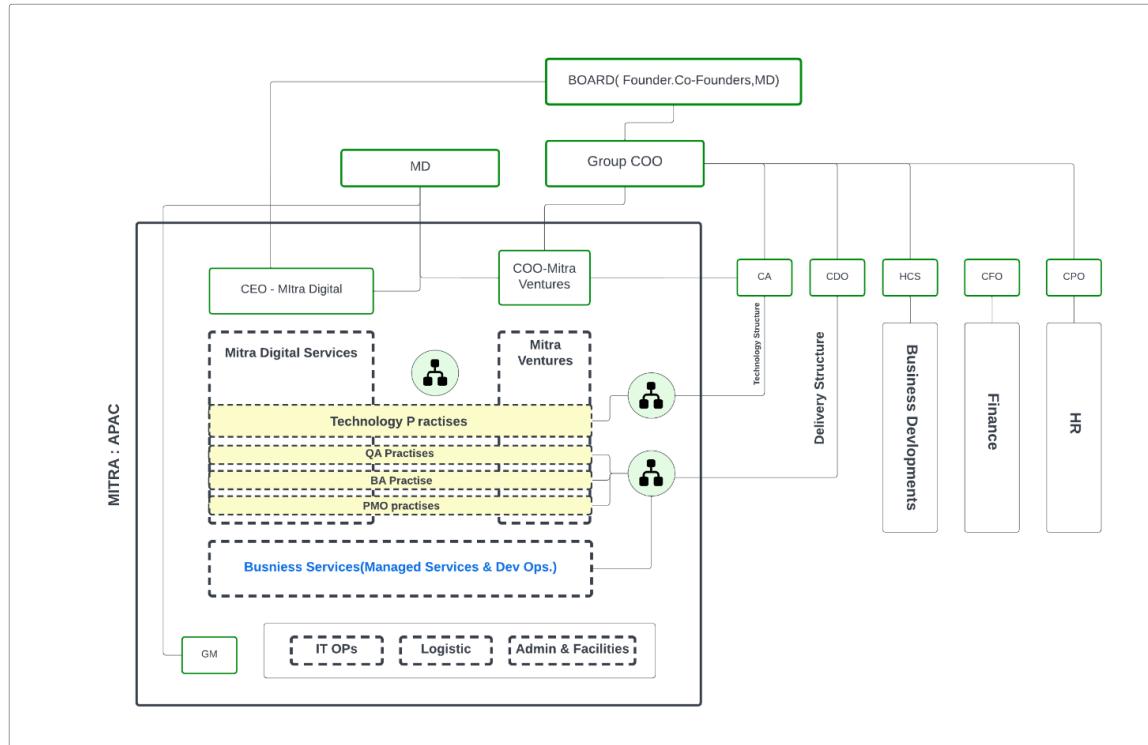


Figure 4-Context of Mitra Innovation

1.2. Recruitment Procedures

Once a vacancy occurs, a detailed job description is prepared before each post is advertised and available to all applicants. All the unfilled vacancies are publicly advertised in local newspapers, job markets, and websites. All the advertisements include the company name and job title. Job description, required qualifications & experience, how to apply, how to get further information, and the closing date for application. On the other hand, Mira innovation has a referral method. Anyone inside the Company can send suitable CVS to the HR department. So, if one of those is selected, he will get a chance to spend a night at a luxury hotel.

When an application is received by the company, the shortlisting panel will review the application and will be shortlisted according to their preference. After that, they will continue the following process for shortlisted applicants.

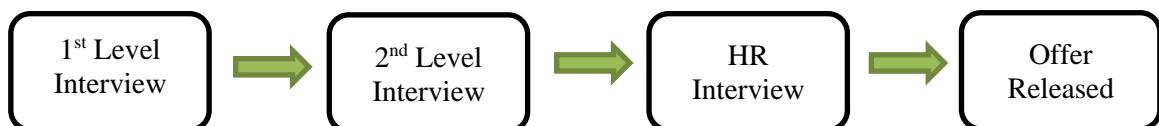


Figure 5-recruitment process stages

If someone was recruited, He/She should have to submit the Mitra Mutual Non-disclosure agreement, Acceptable usage Policy, and Internship agreement.

1.3. Absence / Leave

Mitra recognizes that their employees have a life outside work, and their policies on leave are designed to provide a balanced approach to time off under a range of circumstances, complimenting your annual holiday entitlement. A work-life balance that works for Mitra and its employees is their goal, and they continuously strive for that. All the employees should inform the leaves taken by them to the H.R. department of the company through the supervisor or the project manager.

1.3.1. Annual Leave

The holiday year runs from 1st January to 31st December. Details of every employee's annual holiday entitlement are provided in their Contract of Employment. Mitra policy is that we do not allow carry forward of A/L from previous years unless previously agreed with your line manager/senior management and for exceptional reasons

1.3.2. Other Leave

The company is committed to creating a healthy and positive environment at work where all employees can give their best. This means that we work with our people to reasonably balance the demands of work and private life. In certain circumstances, we recognize there may be a need to take time off to deal with urgent domestic issues. In such circumstances, paid or unpaid compassionate leave or dependents leave may be granted, and we have to discuss the matter with your H.R. Business Partner

1.4. Pay and Benefits

Salary progression is with annual reviews conducted annually in April and October in Sri Lanka and (March in the UK), any increase is at the discretion of the company. Pay details are included in your Contract of Employment. Every employee will be paid monthly in arrears on or around the 26th of the month in the UK (28th in Sri Lanka). Your hours and place of work are included in your Contract of Employment. It is a condition of your employment that you work such necessary and reasonable hours to complete your duties. The company operates a pension scheme, and you will have the opportunity to join the scheme or will be auto-enrolled if you are a qualified joiner.

1.5. SWOT Analysis

1.5.1. Strengths

Mitra innovation is a leading technology company, enabling clients to embrace digital solutions to transform their business. With clients spanning six core sectors, including finance and telecommunications, Mitra provides many services, from system integration and cloud migration to managed services and application development.

1.5.2. Weaknesses

Due to the current economic crisis in Sri Lanka, the I.T. sector has faced lots of problems. Most of them were affected due to the power cut, the team performance decreased, and they were not able to meet the deadlines. But Mitra innovation's top management reduces these types of risks by giving power generators and U.P.S. to their permanent workers, and they have hired lots of hotels and many buildings equipped with high-powered generators that can even face 24-hour power cuts.

The next thing is I have observed that they have no proper plans for new recruitments to the company. Because they prioritize the capacity plan for their new recruitment. Even the worst case is that they recruited people after starting the project. As a result, it may affect billing and loss for the company side. Because, for some reason, projects may be challenging, and sudden recruitment may not be worth the investment for the project and for the company. Therefore, company's HR department should set the most suitable persons for each project prior to starting the development. Because if they cannot find suitable people for the project at the start, the whole team will get burnout, and the project will crash in the middle. Therefore, the company should think about managing its most experienced people in projects wisely. On the other hand, the company should consider maintaining a healthy work-life balance among the developers in every team. Because healthy and happy employees are always the key factor to company success.

From the project management side also, some areas should be developed. Because each developer's capabilities might be different. Therefore, while making the timeline, the person who is deciding the timeline of the project should talk with the respective developer. Otherwise, it will create unnecessary pressure while working on the project.

The next weakness was that the company had no pre-planned Internship programs for government university undergraduates. Therefore, it is good to create some workshops and

compulsory courses at Mitra Innovation because every engineer who passes out from government universities is more knowledgeable than the private sector students. On the other hand, the company should think about Why WSO2 and L.S.E.G. kind of companies are seeking engineering undergraduates and how they build up engineers inside their companies. Because increasing only the no of working heads in the company will not help to speed up the company's success.

1.5.3. Opportunities

In Mitra innovation, they hired different types of engineers. As an example, both computer engineers and electronic/electrical engineering are working inside Mitra innovation. Therefore, there are many opportunities to work in Mitra innovation. Hence Mitra innovation works with foreign clients all the permanent employees get the chance to peg their salaries.

1.5.4. Threats

One of the threats to Mitra Innovation is they don't expand their infrastructure facilities. For example, Mitra doesn't have company vehicles to give transport facilities to their employees who are willing to come to the office and work. Even they didn't try to build more permanent places around Colombo. Hiring buildings would not be a key factor; they are temporary assets for the company. Hence Now Mitra is well set, they have to look to the future and expand their facilities within Sri Lanka. Otherwise, they will have to spend huge amounts of money on their building rent fees with this economic crisis, and if they are going to buy assets, then it will be more costly than today. The next thing is Mitra Innovation's employee's work-life balance is not good at all times. I didn't see any entertainment events organized for their employees within this six-month period. For this reason, most of the employees will leave because there are lots of companies that maintain work-life balance well for their employees.

Chapter 2 - Training Experience

2.1. Short description of my first project (Triton)

2.1.1. Introduction

DygiSec products and services built on the Triton platform leverage its AI-powered analytics and automation engines curated threat intelligence and comprehensive attack-vector coverage to offer your team's specialized solutions for extended detection and response and vulnerability management. Triton helps to automate vulnerability management with a cloud-based, machine learning-powered solution. Intelligently prioritize remediation efforts based on actionable recommendations that reflect the context of the client environment. A cloud may connect and disconnect from many networks over the day, and this dynamic nature makes it powerful and difficult to secure. Therefore, Triton is a Cloud Security Posture Management (C.S.P.M.) tool that will be developed with the intention of helping customers secure their cloud environment

by automating security to reduce that attack surface, fix misconfigurations, resolve configuration drift, block unwanted behavior, and keep threats at bay.

2.1.2.Stakeholders

In the first project, there were two parties. They were –

Table 2-Stakeholders details of the Triton project

Mitra stakeholders	Shimantha Perera(Project Manager) ,Thilina Herath(Architect), Deleema Fernando(Tech lead), Reshan Perera (Consultant Engineer),Tharindu Madushanka(S.S.E. (Angular)),Dinura Marsinghe(Q.A.), Kithsiri Ekanayake(UX/UI Designer),Purni Gunawardana(Business Analyst),Tharun Krishnamurthy (Associate S.E.), Saduni Maduhansani(Associate S.E.), Pasindu uduwela(Intern S.E.) , Kavish Rajakaruna(Intern S.E.)
Dygisec Stakeholder	Stefano Harding(Product Owner), Lakshitha Herath(co-owner), Chanditha Samaranayake(Project Sponsor), Rajesh Rasiah(Risk & compliance)

2.1.3.Tools and techniques

Table 3-Tools and techniques used in project

Jira	This collaborative tool was used to plan, assign, track, report and manage project-related work.
Confluence	This collaborative tool was used to maintain project-related documentation
Slack	This collaborative tool was used for project-related communication among the Mitra and Dygisec teams.
Lucid Chart	This collaborative tool was used to design flow charts and mind maps in order to identify system functionalities better.
Invision/Figma	This collaborative freehand tool was used to design wireframes.

In the project, each developer was assigned tasks using the JIRA task board by the project manager. In the beginning project manager creates the task and assigns the developer and reporter for the tasks. So, you have to update the working time using JIRA. But in the end, due to the project complexity, we have to create a new task for ourselves to track the individual work done by each developer.

In the faculty, we were not familiar with the git commands. However, I was familiarized with the git commands while working on the project. For that, I have to follow some basic tutorials and practice them using my personal GitHub account.

2.1.4.Knowledge regarding GIT

First, I download the GIT installer from its official website. After that, I installed and set GIT bash to execute GIT commands. Then I gave my GitHub credentials to set up GIT for the company email. The commands used for that purpose can be shown below.

- git config --global user.name "github_username"
- git config --global user.email "email_address"

I have to use the VSCode code GitHub extension in Triton because the back-end repo was cloned into WSL.

2.1.4.1. Basic GIT operations used in the project

Table 4-Git commands Used in the Project

commands	operations
git clone <https://name-of-the-repository-link>	Cloned the project from GitHub to local workspace
git checkout -b <branch>	Used to make a new branch
git branch -list	Used to view all the branches
git branch -d <branch-name>	Used to delete a branch
git checkout <branch>	Used to jump to a required branch
git status	Used to get details related to the current branch such as branch update date, to be commits be pushed and to be pulled.
git commit -m "commit message"	Used this command to push the code to a remote GitHub repository using a commit message
git push origin	Push to the main branch of the remote GitHub repository
git pull origin	Fetch the current workings from the remote main branch and merge them with the local branch where you execute the code
git merge	Merge a branch with another branch . It might be developed or the main branch in the remote GitHub repository.
git branch -a	List all the remote branches

2.2. Tasks Assigned in the First project

2.2.1.Task 1- Golang REST API development for compliance dashboard

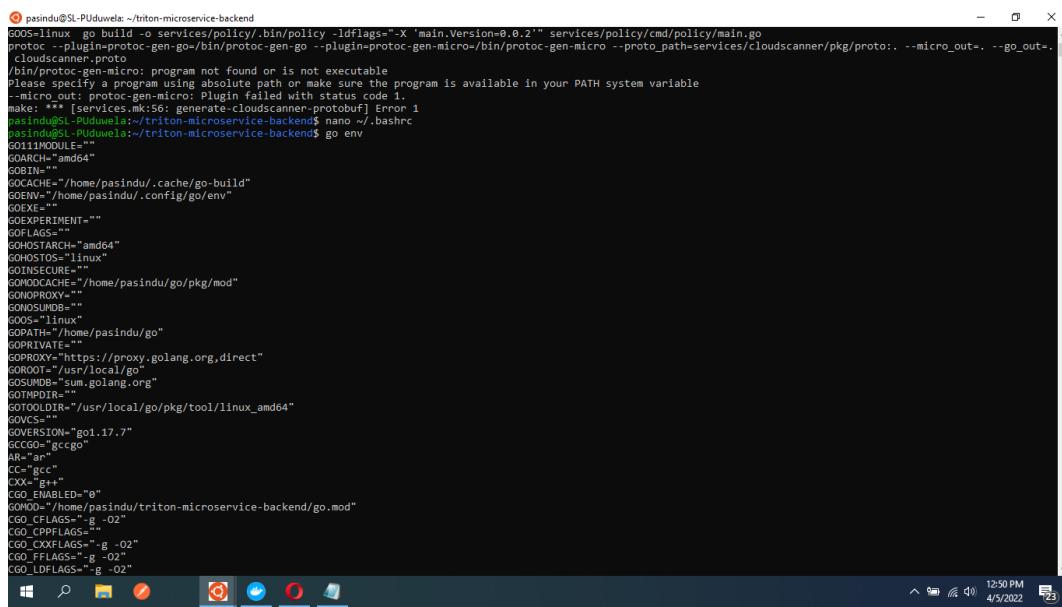
• Introduction to the Task

In this task, I have to create API endpoints for the compliance dashboard service to retrieve some hard quoted data from the back-end.

• Background Knowledge Gathering

When we onboarded into the project, the back-end developments were initially started, and some of the developments were there. I must first clone the project repository from the GitHub repository "digisecure/triton-microservices-backend." Hence all the developments were done inside the WSL. Therefore, first I installed the docker desktop and WSL ubuntu terminal 20.04.4. After completing that, the back-end repository was cloned inside the WSL environment. But without environment configuration, the back-end repository wasn't working on my laptop. For that, I have to install Golang 1.17.7, the latest version at that moment. Then I added the Go path to the environment variable. After completing those steps, I checked the go version because I found that the project is only working for Golang 1.17.7 version. Following are the commands I used for the above steps to configure the back-end Golang working environment.

- ✓ `$ rm -rf /usr/local/go && tar -C /usr/local -xzf go1.17.7.linux-amd64.tar.gz`
- ✓ `export PATH=$PATH:/usr/local/go/bin`
- ✓ `go version or go env`



```
pasindu@SL-PUDuwela: ~/triton-microservice-backend
$ GOOS=linux go build -o services/policy/bin/policy -ldflags="-X main.Version=0.0.2" services/policy/cmd/policy/main.go
proto --plugin=protoc-gen-go/bin/protoc-gen-go -plugin=protoc-gen-micro/bin/protoc-gen-micro --proto_path=services/cloudscanner/pkg/proto: --micro_out=. --go_out=.
/bin/protoc-gen-micro: program not found on is not executable
Please specify the program using absolute path or make sure the program is available in your PATH system variable
micro: failed to run program: exec: "protoc-gen-micro": executable file not found in $PATH
make: *** [services.mk:56: generate-cloudscanner-protobuf] Error 1
pasindu@SL-PUDuwela:~/triton-microservice-backend$ nano ~/.bashrc
pasindu@SL-PUDuwela:~/triton-microservice-backend$ g env
GO111MODULE=""
GOARCH="amd64"
GOBIN=""
GOCACHE="/home/pasindu/.cache/go-build"
GOECDL="/home/pasindu/.config/go/env"
GOEXE=""
GOEXPERIMENT=""
GOFLAGS=""
GOHOSTARCH="amd64"
GOHOSTOS="linux"
GOINSECURE=""
GOMODCACHE="/home/pasindu/go/pkg/mod"
GONOPROXY=""
GONOSUMDB=""
GOOS="linux"
GOPATH="/home/pasindu/go"
GOPRIVATE=""
GOPROXY="https://proxy.golang.org,direct"
GOROOT="/usr/local/go"
GOSUMDB="sum.golang.org"
GOTMPDIR=""
GOTOOLDIR="/usr/local/go/pkg/tool/linux_amd64"
GOVCS=""
GOVERSION="go1.17.7"
GCCGO="gccgo"
AR="ar"
CC="gcc"
CXX="g++"
CGO_ENABLED="0"
GOMOD="/home/pasindu/triton-microservice-backend/go.mod"
CGO_CFLAGS="-g -O2"
CGO_CPPFLAGS=""
CGO_CXXFLAGS="-g -O2"
CGO_FFLAGS="-g -O2"
CGO_LDFLAGS="-g -O2"
Windows PowerShell
Version 7.0.10240 Build 10240
Copyright (C) Microsoft Corporation. All rights reserved.

12:50 PM 4/5/2022
```

Figure 6-Check the environment variables set in WSL

After that, if not the vs code IDE detect the working environment, I have to install some vscode extensions such as remote container, WSL, Github login, Docker, Go, Angular and Kubernetes extensions from the VSCode IDE. After installing all the extensions, VSCode will identify the working environment as WSL, as in figure 4. The docker extensions would help to sync with the docker desktop and build the images.

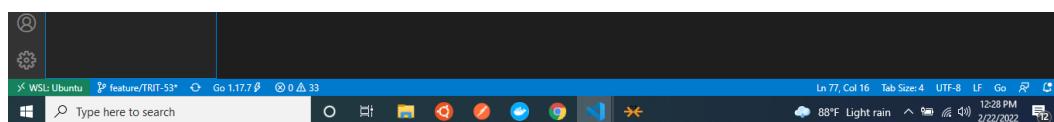


Figure 7-After installing the necessary extensions in VSCode

In the beginning, I have to clone the front-end repository also. Because I have to be familiar with both the front end and back-end. For that, I have to install angular CLI and necessary libraries inside the WSL environment. After installing angular CLI and other libraries inside WSL, the front-end repository builds successfully.

```
asindu@SL-PUduwela:~/triton-frontend-portal$ npm version
{
  "ngx-admin": "8.0.0",
  "npm": "6.14.16",
  "ares": "1.18.1",
  "brotli": "1.0.9",
  "cldr": "40.0",
  "icu": "70.1",
  "llhttp": "2.1.4",
  "modules": "83",
  "napi": "8",
  "nghttp2": "1.42.0",
  "node": "14.19.0",
  "openssl": "1.1.1m",
  "tz": "2021a3",
  "unicode": "14.0",
  "uv": "1.42.0",
  "v8": "8.4.371.23-node.85",
  "zlib": "1.2.11"
}
asindu@SL-PUduwela:~/triton-frontend-portal$ npm run build
+ ngx-admin@8.0.0 build /home/pasindu/triton-frontend-portal
  ng build
  ...
  Generating browser application bundles (phase: setup)...
```

Figure 8-Successful build of the frontend repository

• Design of the component

In the project microservice architecture was used. There was no designing part for me because the client-side senior architect already did the Microservice back-end component design. We have to follow the design pattern and continue coding in the back-end.

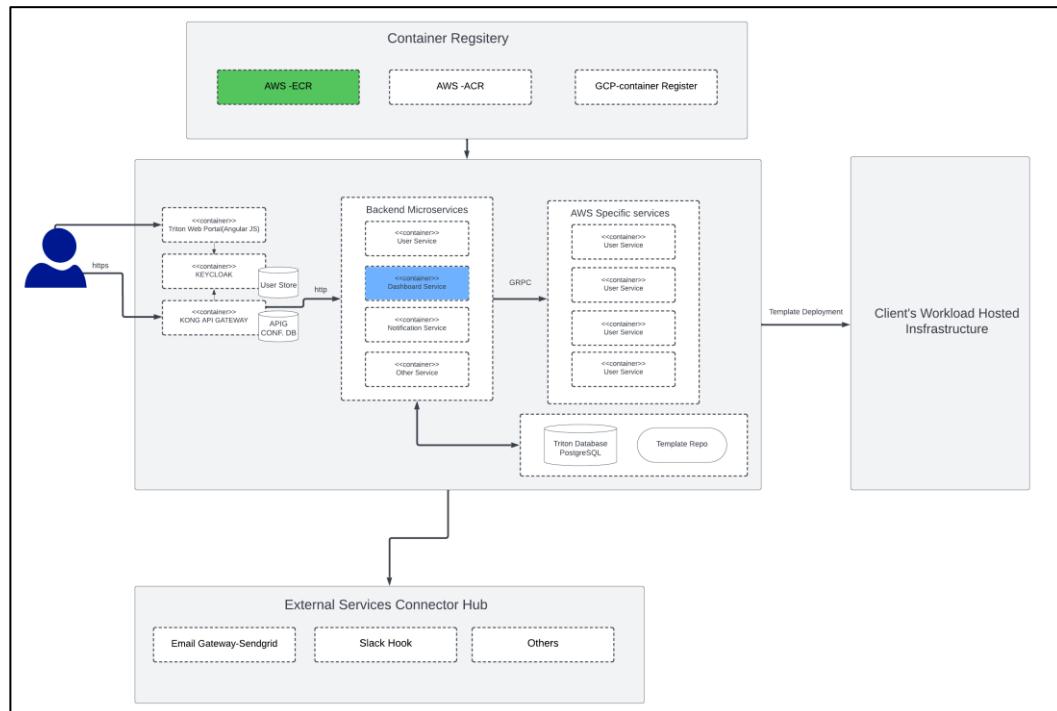


Figure 9-Microservice Backend component design Structure

- **Implementation and Testing**

The implementation phase was not tricky; hence I was able to follow the pattern of writing Golang REST APIS. After creating the Endpoint successfully, I have to test it in the local environment using the Postman tool. However, to test the work first, I have to build the project without any errors. For that, I have to spend more time resolving the issues. Finally, the issues were solved, and I was able to access the API endpoint through the browser and postman. After every successful complete build, as below in Figures 8 &9, all the pods should be in a running state, and the services also should be assigned by clusterIP and external port. The build takes 10 to 15 minutes, according to the Internet speed.

```

pasindu@SL-PUDuwela: ~/triton-microservice-backend$ replicaset.apps/keycloak-8cb98549f      1/1   1     0     2m
replicaset.apps/policy-service-8479bfff85     1/1   1     1     119s
replicaset.apps/portal-5df47474fd            1/1   1     1     119s
replicaset.apps/triton-gateway-kong-798ffbf9b9 1/1   1     0     2m2s
replicaset.apps/user-service-5b9474579d        1/1   1     1     119s

NAME          READY   STATUS    RESTARTS   AGE
statefulset.apps/triton-cloudscanner-db      1/1   Running   0          2m29s
statefulset.apps/triton-db-postgresql       1/1   Running   0          2m22s
pasindu@SL-PUDuwela:~/triton-microservice-backend$ make app-info
Kubectl get all -n triton-dev
NAME          READY   STATUS    RESTARTS   AGE
pod/cloudconnect-service-5ed54fbcb6-omfm    1/1   Running   0          2m22s
pod/cloudscanner-aws-service-6b7555b88-qzltq 1/1   Running   0          2m22s
pod/cloudscanner-service-7cd6795b4d-xbrrt   1/1   Running   2 (2m16s ago) 2m22s
pod/dashboard-service-7954665640-47xs2      1/1   Running   0          2m23s
pod/keycloak-54474579d-92dij                1/1   ErrImagePull 0          2m23s
pod/policy-service-8479bfff85-92dij         1/1   Running   0          2m23s
pod/portal-5df47474fd-674gv                 1/1   Running   0          2m23s
pod/triton-cloudscanner-db-0                  1/1   Running   0          2m53s
pod/triton-db-postgresql-0                   1/1   Running   0          2m56s
pod/triton-gateway-kong-798ffbf9b9-8dv5p   2/2   Running   2 (42s ago) 2m26s
pod/user-service-5b9474579d-62x2xm          1/1   Running   2 (2m18s ago) 2m23s

NAME          TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)           AGE
service/cloudconnect-service ClusterIP  10.108.220.199  <none>           8080/TCP          2m22s
service/cloudscanner-service  ClusterIP  10.106.65.75    <none>           8080/TCP          2m22s
service/dashboard-service   ClusterIP  10.109.242.207  <none>           8080/TCP          2m23s
service/keycloak             ClusterIP  10.108.168.101  <none>           8443/TCP, 8080/TCP 2m24s
service/portal-service       ClusterIP  10.111.154.65   <none>           8080/TCP          2m23s
service/portal               ClusterIP  10.109.17.73    <none>           80/TCP            2m23s
service/triton-cloudscanner-db ClusterIP  10.96.33.15    <none>           7687/TCP, 7474/TCP, 7473/TCP 2m54s
service/triton-cloudscanner-db-admin ClusterIP  10.107.109.215 <none>           6362/TCP, 7687/TCP, 7474/TCP, 7473/TCP 2m54s
service/triton-cloudscanner-db-neo4j LoadBalancer 10.103.63.98  127.0.0.1  7474:30604/TCP, 7473:31935/TCP, 7687:32690/TCP 2m53s
service/triton-db-postgresql ClusterIP  None           <none>           5432/TCP          2m56s
service/triton-db-postgresql-hl ClusterIP  None           <none>           5432/TCP          2m56s
service/triton-gateway-kong-proxy LoadBalancer 10.100.54.53  127.0.0.1  80:32317/TCP, 443:32118/TCP 2m25s
service/user-service          ClusterIP  10.97.105.183  <none>           8080/TCP          2m23s

NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/cloudconnect-service  1/1   1           1           2m22s
deployment.apps/cloudscanner-aws-service 1/1   1           1           2m22s
deployment.apps/cloudscanner-service   1/1   1           1           2m22s
deployment.apps/dashboard-service      1/1   1           1           2m23s

```

Figure 10-The Successful build of the backend (Pod & services details)

```

pasindu@SL-PUDuwela: ~/triton-microservice-backend$ replicaset.apps/triton-cloudscanner-db-neo4j LoadBalancer 10.102.239.240 127.0.0.1 7474:32659/TCP,7473:32549/TCP,7687:30842/TCP 4h56m
service/triton-db-postgresql ClusterIP 10.99.247.182 <none> 5432/TCP 4h57m
service/triton-db-postgresql-hl ClusterIP None           <none> 5432/TCP 4h57m
service/triton-gateway-kong-proxy LoadBalancer 10.111.76.44 127.0.0.1 80:32317/TCP,443:32118/TCP 4h56m
service/user-service ClusterIP 10.102.105.81 <none> 8080/TCP 4h56m

NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/cloudconnect-service  1/1   1           1           4h56m
deployment.apps/cloudscanner-aws-service 1/1   1           1           4h56m
deployment.apps/cloudscanner-service   1/1   1           1           4h56m
deployment.apps/dashboard-service      1/1   1           1           4h56m
deployment.apps/keycloak              0/1   1           0           4h56m
deployment.apps/policy-service        0/1   1           1           4h56m
deployment.apps/portal                0/1   1           0           4h56m
deployment.apps/triton-gateway-kong  0/1   1           0           4h56m
deployment.apps/user-service          1/1   1           1           4h56m

NAME          DESIRED  CURRENT  READY   AGE
replicaset.apps/cloudconnect-service-774c4bf667 1       1       1       4h56m
replicaset.apps/cloudscanner-aws-service-6546b08fd9 1       1       1       4h56m
replicaset.apps/cloudscanner-service-bc9ccb9b9 1       1       1       4h56m
replicaset.apps/dashboard-service-66645c796c 1       1       1       4h56m
replicaset.apps/keycloak-5f9bd9b5b 1       1       0       4h56m
replicaset.apps/policy-service-cbfb8f75d 1       1       1       4h56m
replicaset.apps/portal-c444fcdb 1       1       0       4h56m
replicaset.apps/triton-gateway-kong-75bd6f7dc8 1       1       0       4h56m
replicaset.apps/user-service-789675799 1       1       1       4h56m

NAME          READY   AGE
statefulset.apps/triton-cloudscanner-db 1/1   4h56m
statefulset.apps/triton-db-postgresql 1/1   4h57m

NAME          COMPLETIONS  DURATION   AGE
job.batch/triton-gateway-kong-init-migrations 0/1   4h56m  4h56m
pasindu@SL-PUDuwela:~/triton-microservice-backend$ 

```

Figure 11- The sucessfull build of the backend(replicaset & deployments details)

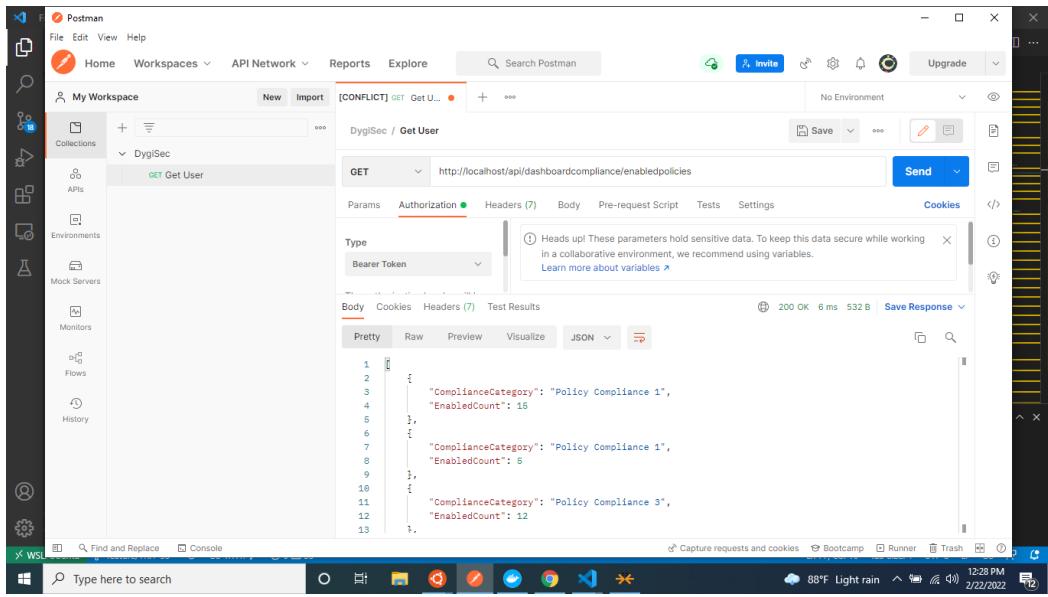


Figure 12-Testing the API endpoints using Postman IDE

• Problems Faced during Implementation and Resolutions

At the project cloning, many errors occurred due to misconfigurations done inside the WSL. Therefore have to spend more time and solve the issues one by one. I have to sync up with another developer and follow this coding pattern.

```

pasindu@SL-PUDuwela:~/triton-microservice-backend$ docker build -f ./deployment/db/Dockerfile . / deployment/db -t triton-postgres:0.0.1
[+] Building 0.6s (3/3) FINISHED
--> [internal] load build definition from Dockerfile          0.1s
--> [internal] transfering dockerfile: 37B                   0.0s
--> [internal] load .dockerignore                           0.1s
--> [internal] transferring context: 2B                  0.0s
--> ERROR [internal] load metadata for docker.io/bitnami/postgresql:14.2.0      0.0s
--> [internal] load metadata for docker.io/bitnami/postgresql:14.2.0:
-----
Failed to solve with frontend dockerfile.vb: failed to create LLB definition: failed to do request: Head "https://registry-1.docker.io/v2/bitnami/postgresql/manifests/14.2.0": dial tcp: lookup registry-1.docker.io on 192.168.65.5:53: no such host
make: *** [common.mk:31: build-db-image] Error 1
pasindu@SL-PUDuwela:~/triton-microservice-backend$ 

```

Figure 13-Due to poor Internet connection Some docker files were failed to build

- **Lessons learned and experience gained**

All the APIs I created for the compliance dashboard used the GET method. But I have the chance to learn about the POST, DELETE, and PUT type API endpoints. I learned about the benefits and drawbacks of microservice architecture. The next thing was that I was able to work with the newest technologies such as docker and Kubernetes.

2.2.2.Task 2 – Swagger API documentation for Services

- **Introduction to the Task**

All REST API endpoints in the project should be documented using Swagger UI. The primary purpose of building swagger UI was to document all the API Endpoints in service in one place. For future developments, swagger API documentation will be beneficial.

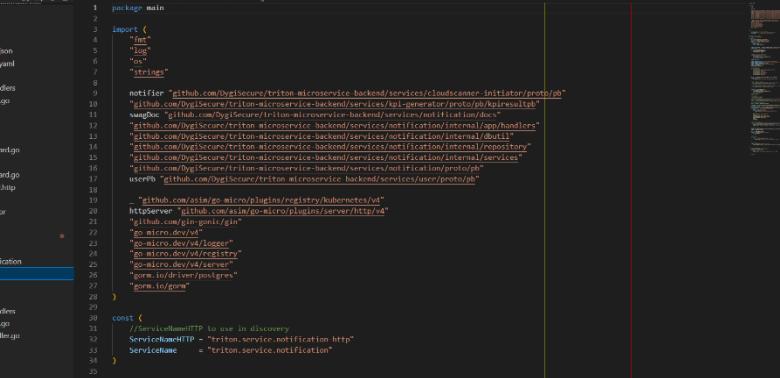
- **Background Knowledge Gathering**

At the task, beginning swagger UI was new to me; therefore, I followed some video tutorials to become familiar with the swagger API documentation process. As a sub-task, First, I have to identify a suitable library for the swagger API documentation. Hence we are using the Gin web framework, I have to search for a library that is supported by the Gin web framework. Finally, I have proposed the swaggo/swag library for swagger UI implementation.

- **Design of the component**

After naming the library, I followed swaggo/swag library documentation and learned how to convert Go annotations into swagger documentation 2.0. After reading the documentation, I follow the below steps to generate docs.go, swagger.json, and swagger.YAML files. There were some methods to create the swagger UI manually using online web tools. However, for ease of use, I preferred to use swag CLI to generate the files required to create the swagger UI. For that, I have followed the below steps for one service. After successfully loading the swagger UI from the browser, I tested it by the options given from the swagger UI. Then I moved to implement the swagger UI for the other services that were already finished at that moment.

1. First, Import the necessary packages to the handler. go file



The screenshot shows the Visual Studio Code interface with the Go extension installed. The current file is `main.go` located in the `triton-microservice/notification/cmd/notification` directory. The code implements a gRPC service for notifications, defining a package `main` and a service `NotificationService`. It includes imports for `log`, `fmt`, `io`, `os`, and `strings`. The `NotificationService` has methods `Notify` and `GetLogs`. The `Notify` method takes a `NotifyRequest` and returns a `NotifyResponse`. The `GetLogs` method takes a `GetLogsRequest` and returns a `GetLogsResponse`. The code uses `github.com/grpc/grpc-go/grpc` for gRPC and `github.com/golang/protobuf/proto` for protocol buffers. The code editor shows syntax highlighting for Go and JSON files, and the status bar indicates the file is 100% complete.

```
File Edit Selection View Go Run Terminal Help
        manage triton microservice backend develop Visual Studio Code
EXPLORE . . .
TRITON-MICROSERV... main.go
services> notification> cmd> notification> >> main.go ...
1 package main
2
3 import (
4     "fmt"
5     "log"
6     "os"
7     "strings"
8
9     "notifier"
10    "github.com/gyssecure/triton-microservice-backend/services/cloudscanner-initiator/proto"
11    "github.com/gyssecure/triton-microservice-backend/services/gpu-generator/proto"
12    "github.com/gyssecure/triton-microservice-backend/services/notification/docs"
13    "github.com/gyssecure/triton-microservice-backend/services/notification/internal/api"
14    "github.com/gyssecure/triton-microservice-backend/services/notification/internal/grpc"
15    "github.com/gyssecure/triton-microservice-backend/services/notification/internal/services"
16    "github.com/gyssecure/triton-microservice-backend/services/notification/proto"
17    "github.com/gyssecure/triton-microservice-backend/services/user/proto"
18
19    "github.com/asim/go-micro/plugins/registry/kubernetes/v1"
20    "https://github.com/asim/go-micro/plugins/server/http/vd"
21    "github.com/gin-gonic/gin"
22
23    "go-micro.dev/v4/logger"
24    "go-micro.dev/v4/registry"
25    "go-micro.dev/v4/server"
26    "gorm.io/driver/postgres"
27    "gorm.io/gorm"
28
29)
30
31 const {
32     // ServiceNameHTTP = "triton-service-notification"
33     ServiceNameHTTP = "triton.service.notification"
34     ServiceName = "triton.service.notification"
35 }
36
37 // Version of the service
38 Version = "development"
39
40 func main() {
41 }
```

Figure 14- pakages imported into handler.go file

2. Added the General API annotations in main.go file reserved for each service

```
File Edit Selection View Go Run Terminal Help handler.go Triton microservice backend develop Visual Studio Code
EXPLORER TRITON-MICROSERV... ... handler.go
services > notification > internal > app > handlers > handler.go ...
1 package handlers
2
3 import (
4     "net/http"
5     "strconv"
6
7     "github.com/BygSecure/triton-microservice-backend/services/notification/internal/dto"
8     "github.com/BygSecure/triton-microservice-backend/services/notification/internal/services"
9
10    "github.com/gin/gonic/gin"
11
12    swaggerFiles "github.com/swaggo/files" // swagger embed files
13    ginSwagger "github.com/swaggo/gin-swagger" // gin-swagger middleware
14)
15
16 const (
17     XTritonSerial = "X-Triton-IDPSerial"
18 )
19
20 type Handler struct {
21     notificationService services.NotificationService
22     notifierService   services.NotifierService
23 }
24
25 func NewHandler(notificationService services.NotificationService, notifierService services.NotifierService) Handler {
26     return Handler{
27         notificationService: notificationService,
28         notifierService:   notifierService,
29     }
30 }
31
32 func (h Handler) DeleteWithRouter(*gin.Engine) {
33     root := router.Group("/api/internal/notify")
34     root.DELETE("/:id", h.deleteAlert)
35     root.GET("", h.getAlerts)
36     root.GET("/:count", h.getNewestCount)
37 }
38
39 configure := root.Group("/configure")
40 configure.GET("/", h.getActivities)
41
42
```

Figure 15-General API Annotations in main.go file

3. Added the API annotation in the handler.go file

Figure 16-API annotation in handler.go file

After that, I tried different codes and finally found the correct `swag init` command with the correct command options and arguments.

```
② pasindu@SL-Püdüwela:~/triton-microservice-backend
```

```
2022/03/18 09:35:54 Generate general API Info, search dir: ./services/dashboard
[2022/03/18 09:35:54] 2022-03-18T09:35:54: warning: failed to get package name in dir: ./services/dashboard, error: execute go list command, exit status 1, stdout: stderr: no Go files in /home/pasindu/triton-microservice-backend/services/dashboard
[2022/03/18 09:35:54] 2022-03-18T09:35:54: execute go list command, cddir /home/pasindu/triton-microservice-backend/services/dashboard/services/dashboard, error: no such file or directory, stdout: stderr:
```

```
2022/03/18 09:40:22 Generate swagger docs...
2022/03/18 09:42:51 Generate general API Info, search dir: ./services/dashboard
2022/03/18 09:42:52 warning: failed to get package name in dir: ./services/dashboard
[2022/03/18 09:42:52] 2022-03-18T09:42:52: warning: failed to get package name in dir: /home/pasindu/triton-microservice-backend/services/dashboard, error: execute go list command, exit status 1, stdout: stderr: no Go files in /home/pasindu/triton-microservice-backend/services/dashboard
[2022/03/18 09:44:41] 2022-03-18T09:44:41: warning: failed to get package name in dir: ./services/dashboard/cmd/dashboardservice does not exist
[2022/03/18 09:44:41] 2022-03-18T09:44:41: warning: failed to get package name in dir: ./services/dashboard/cmd/dashboardservice does not exist
[2022/03/18 09:45:02] 2022-03-18T09:45:02: warning: failed to get package name in dir: ./services/dashboard/cmd/dashboardservice does not exist
[2022/03/18 09:45:18] 2022-03-18T09:45:18: Generate swagger docs...
2022/03/18 09:45:18 Generate general API Info, search dir: ./services/dashboard/cmd/dashboardservice
```

```
pasindu@SL-Püdüwela:~/triton-microservice-backend$ swag init -d ./services/dashboard/cmd/dashboardservice -o ./docs --parseDependency --parseVendor
```

```
2022/03/18 09:47:47 Generating swagger docs...
2022/03/18 09:47:47 Generate general API Info, search dir: ./services/dashboard/cmd/dashboardservice
[2022/03/18 09:50:07] 2022-03-18T09:50:07: Generating dto.General
[2022/03/18 09:50:07] 2022-03-18T09:50:07: Generating dto.Count
[2022/03/18 09:50:07] 2022-03-18T09:50:07: Generating dto.RiskStatus
[2022/03/18 09:50:07] 2022-03-18T09:50:07: Generating dto.Asset
[2022/03/18 09:50:07] 2022-03-18T09:50:07: Generating dto.TopAssetByRisk
[2022/03/18 09:50:07] 2022-03-18T09:50:07: Generating dto.Data
[2022/03/18 09:50:07] 2022-03-18T09:50:07: Generating dto.HighlyRiskThreatStatus
[2022/03/18 09:50:07] 2022-03-18T09:50:07: Generating dto.Alerts
[2022/03/18 09:50:07] 2022-03-18T09:50:07: Generating dto.Severity
[2022/03/18 09:50:07] 2022-03-18T09:50:07: Generating dto.Category
[2022/03/18 09:50:07] 2022-03-18T09:50:07: Generating dto.CertificationLog
[2022/03/18 09:50:07] 2022-03-18T09:50:07: Generating dto.RiskThreat
[2022/03/18 09:50:07] 2022-03-18T09:50:07: Generating dto.Summary
[2022/03/18 09:50:07] 2022-03-18T09:50:07: Generating dto.ActiveIssues
[2022/03/18 09:50:07] 2022-03-18T09:50:07: Generating dto.PolicyViolations
[2022/03/18 09:50:07] 2022-03-18T09:50:07: Generating dto.SeverityCovverages
[2022/03/18 09:50:07] 2022-03-18T09:50:07: Generating dto.PolicyAlerts
[2022/03/18 09:50:07] 2022-03-18T09:50:07: create docs.go at ./docs/docs.go
[2022/03/18 09:50:07] 2022-03-18T09:50:07: create swagger.yaml at ./docs/swagger.yaml
[2022/03/18 09:50:07] 2022-03-18T09:50:07: create swagger.json at ./docs/swagger.json
```

```
pasindu@SL-Püdüwela:~/triton-microservice-backend$
```

Figure 17-Final Successful swag init command

Likewise, I continued the same work for policy, notification, main, and compliance dashboard services. The swagger UI was able to access using the following pattern URL shown in the figure 16 for all the service

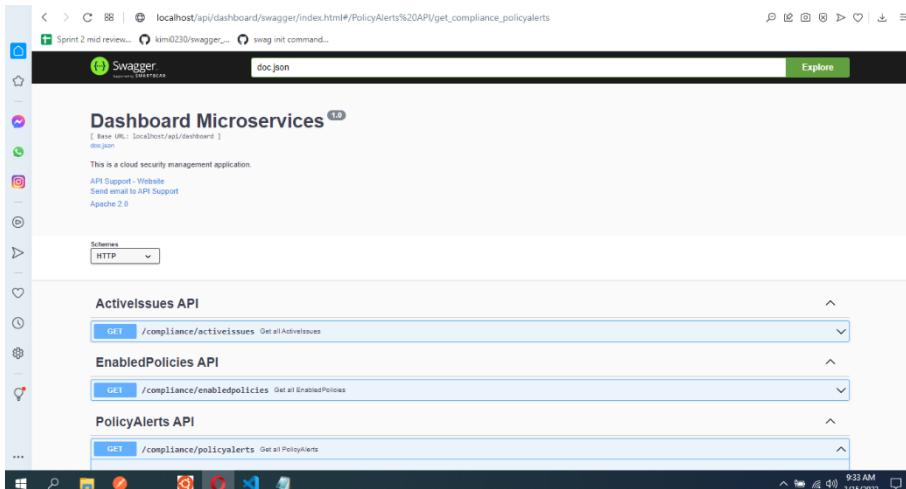


Figure 18- Initial Loading Overview of SwaggerUI

• **Implementation and Testing of the Tool**

From the swagger UI, we can check all the API endpoints.

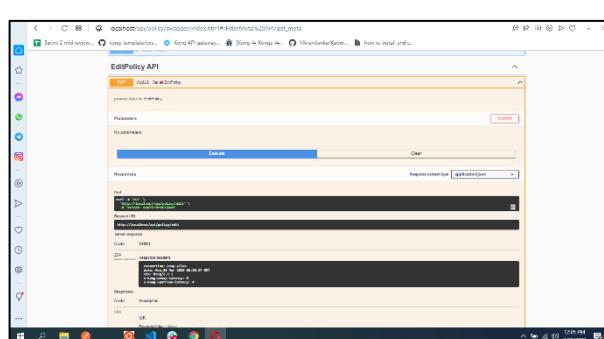


Figure 19-PUT method API Endpoint

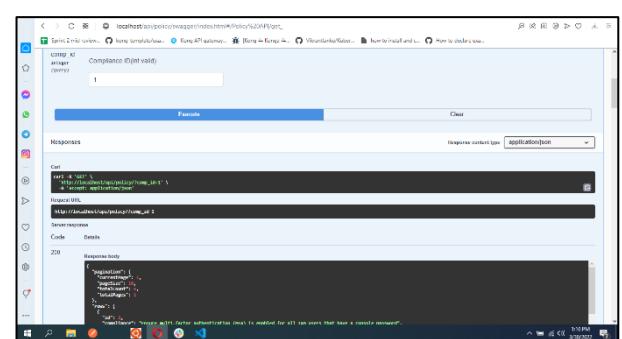


Figure 20-GET method API Endpoint

• Problems Faced during Implementation and Resolutions

Many difficulties were issued while choosing the correct swagger library for the project. Because some of the libraries are inoperable. Therefore, I have to find a library that supports the Gin web framework. The next problem occurred while finding the correct swag init command. It took more time because the project structure was different In Triton, and I was not able to identify the desired path of the swag init command. The tech lead suggested to me, to keep all the swagger environment variables on a single file, display the swagger release version, and check if the domain changes; swagger will open it swagger UI for a new domain without any code change. To check that the swagger UI opens for any domain, I have created a host inside my machine named triton.com. Then I tried the following URL.

URL - <http://triton.com/swagger/api/dashboard>

URL works without any problem for the new host. After deploying the back-end in AWS EKS, swagger also worked without errors. For the environment variable issue, I came up with a solution to use the Golang file reading mechanism to access that information from a separate file. But it was unsuccessful because the file reading wasn't working after building the image. Then I found a solution to set the swagger version and set general info dynamically inside the main function of each service as below.

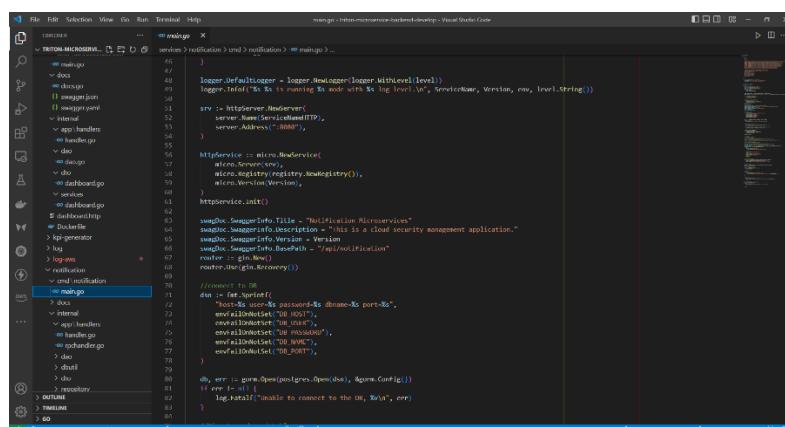


Figure 21-Programmatically set the swagger details

- Lessons learned and experience gained

Here I have learned the importance of having a swagger API documentation. Because we can access each API endpoint using Swagger API documentation. On the other hand, I learned about the status codes such as 201-status Ok, 400-failure, 404- not found,409-conflict and 405- Method not allowed etc.

2.2.3.Task 3 -Done research on creating an O.I.D.C. plugin for the Kong gateway

• Introduction to the Task

I was assigned to research to build an OpenId connect & OAuth2.0 plugin for the Kong gateway.

• Background Knowledge Gathering

I have to work with keycloak and the kong gateway on this task. For that, I have followed some tutorials related to kong gateway and keycloak. Below is some information that I found on Kong Gateway and Keycloak.

Table 5-Details On Kong Gateway, Keycloak, and Microservice architecture

Layer	Application Component	version	Service responsibility
API Management	Kong API Gateway	2.7.0	<ul style="list-style-type: none">• Provide Open APIs for internal consumers.• Portal for API creators. API Lifecycle will be managed here• Portal for API discovery and subscription• API Rate limiting and throttling
Identity Management	Keycloak Identity server	16.1.1	<ul style="list-style-type: none">• Centralized Identity Management Solution• Manage all the identities of the system users• Manage MFA (multi-Factor Authentication) to enhance the security
Microservices	Kubernetes Platform	1.23	<ul style="list-style-type: none">• Provide business logic-related microservices applications• Domain Models• Event sourcing capabilities for future extensions.
	Container Service	AWS ECR	<ul style="list-style-type: none">• Host all the microservices docker images

• Design of the component

I have proposed to the tech lead two ways that we can implement OIDC and OAuth2.0 plugins in Triton. For the first method, I have proposed the below workflow diagram with Another developer suggested on his blog site.

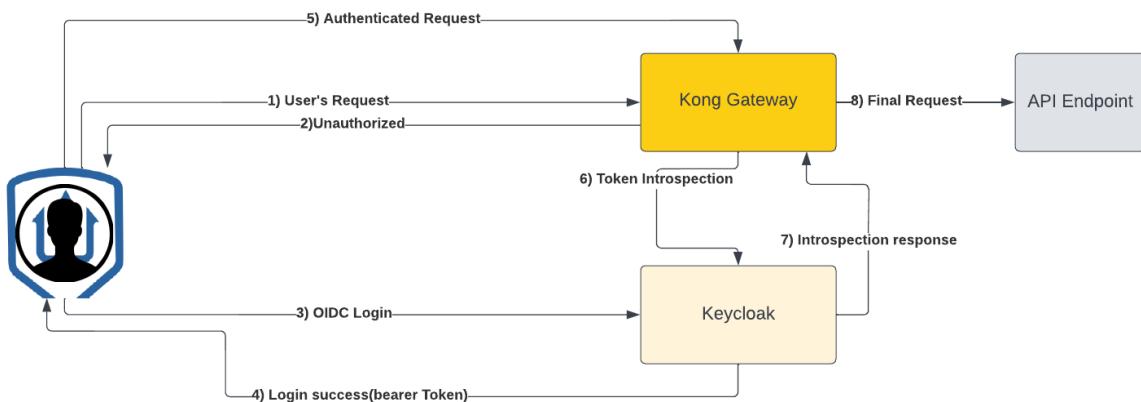


Figure 22- Proposed Workflow of the plugin

As the second method, I have suggested using the Konga dashboard instead of creating the whole plugin at the beginning. Konga is a fully-featured open-source, multi User GUI that makes the hard task of managing multiple kong installations easy way. It can be integrated with MySQL, PostgreSQL, and MongoDB databases and provides the GUI for better understanding and maintaining the architecture. Through Konga, we have the chance to activate free plugins. But it was risky and not the best solution for the situation.

The screenshot shows the Konga dashboard interface. The left sidebar contains navigation links for Dashboard, API GATEWAY, SERVICES, ROUTES, PLUGINS, UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHTOS, and SETTINGS. The main content area includes a summary of connections (ACTIVE: 3, READING: 0, WRITING: 2, WAITING: 1, ACCEPTED: 38K+, HANDLED: 38K+) and total requests (40K+). Below this are sections for NODE INFO (HostName: ip-10-0-0-242, Tag Line: Welcome to kong, Version: 0.31, LUA Version: LuaJIT 2.1.0-beta3, Admin listen: ["0.0.0.0:8001", "0.0.0.0:8444 ssl"]), TIMERS (Pending: 1, Running: 0), and DATASTORE INFO (DBMS: postgres, Host: kp14fnnbbxz.chr69rj7wah.eu-west-1.rds.amazonaws.com, Database: kong, User: Jp2GxIEJt9p, Port: 5432). At the bottom, there is a list of available PLUGINS: response-transformer, correlation-id, statsd, jwt, cors, basic-auth, key-auth, ldap-auth, http-log, oauth2, hmac-auth, acl, datadog, tcp-log, ip-restriction, request-transformer, file-log, bot-detection, loggly, request-size-limiting, syslog, udp-log, response-rate-limiting, aws-lambda, runscope, rate-limiting, and request-termination.

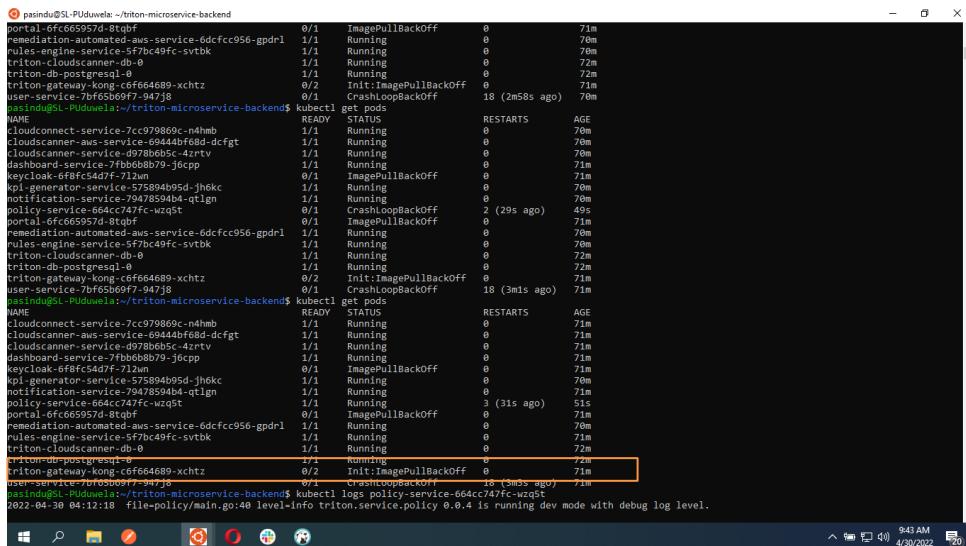
Figure 23- The Overview of Konga dashboard

• Implementation and Testing of the Tool

The implementation wasn't successful; hence the project structure was complex and not expert in Kong API Gateway development .But I have tried to use the Konga dashboard in a separate project and investigate its features. On the other hand, I have tried some blog posts to create a simple plugin on Kong gateway. Following Those steps wasn't successful for the triton project.

• Problems Faced during Implementation and Resolutions

The main reason was every time I tried to add a simple plugin to the current triton kong gateway, it crashed. I tried to read the Kong official documentation to solve the issues, but it was unsuccessful. Therefore I have to leave the task and go for my next assigned task.



```
basindu@SL-PUDuwe1:~/triton-microservice-backend$ kubectl get pods
NAME                               READY   STATUS        RESTARTS   AGE
cloudconnect-service-7cc979869c-n4hmb   1/1    Running      0          78m
cloudscanner-aws-service-694440f68d-dcfgt   1/1    Running      0          78m
triton-cloudscanner-db-0                1/1    Running      0          72m
triton-db-postgresql-0                 1/1    Running      0          72m
triton-gateway-kong-c6f664689-xchtz     0/2    Init:ImagePullBackOff  0          71m
user-service-7bf65b697-947j8            0/1    CrashLoopBackOff  18 (2m58s ago)  70m
basindu@SL-PUDuwe1:~/triton-microservice-backend$ kubectl get pods
NAME                               READY   STATUS        RESTARTS   AGE
cloudconnect-service-7cc979869c-n4hmb   1/1    Running      0          78m
cloudscanner-service-d9780b65c-4zrtv     1/1    Running      0          78m
dashboard-service-7fb6b8b79-j6cpp       1/1    Running      0          71m
keycloak-6f8fc54d7f-712an              0/1    ImagePullBackOff  0          71m
kpi-generator-service-575894b95d-jh0kc   1/1    Running      0          78m
notification-service-644cc747894b4d-rlgn  1/1    Running      0          49s
policy-service-644cc747fc-wzqst         0/1    CrashLoopBackOff  2 (29s ago)  49s
portal-6fc665057d-8tbf                0/1    ImagePullBackOff  0          71m
remediation-automated-aws-service-6dcfcc056-gpdr1  1/1    Running      0          70m
rules-engine-service-5f7bc49fc-svtbk    1/1    Running      0          78m
triton-cloudscanner-db-0                1/1    Running      0          72m
triton-db-postgresql-0                 1/1    Running      0          72m
triton-gateway-kong-c6f664689-xchtz     0/2    Init:ImagePullBackOff  0          71m
user-service-7bf65b697-947j8            0/1    CrashLoopBackOff  18 (3m1s ago)  71m
basindu@SL-PUDuwe1:~/triton-microservice-backend$ kubectl get pods
NAME                               READY   STATUS        RESTARTS   AGE
cloudconnect-service-7cc979869c-n4hmb   1/1    Running      0          78m
cloudscanner-service-d9780b65c-4zrtv     1/1    Running      0          78m
dashboard-service-7fb6b8b79-j6cpp       1/1    Running      0          71m
keycloak-6f8fc54d7f-712an              0/1    ImagePullBackOff  0          71m
kpi-generator-service-575894b95d-jh0kc   1/1    Running      0          78m
notification-service-644cc747894b4d-rlgn  1/1    Running      0          51s
policy-service-644cc747fc-wzqst         1/1    Running      3 (31s ago)  51s
portal-6fc665057d-8tbf                0/1    ImagePullBackOff  0          71m
remediation-automated-aws-service-6dcfcc056-gpdr1  1/1    Running      0          70m
rules-engine-service-5f7bc49fc-svtbk    1/1    Running      0          78m
triton-cloudscanner-db-0                1/1    Running      0          72m
triton-gateway-kong-c6f664689-xchtz     0/2    ImagePullBackOff  0          72m
triton-gateway-kong-c6f664689-xchtz     0/2    Init:ImagePullBackOff  0          71m
user-service-7bf65b697-947j8            0/1    CrashLoopBackOff  18 (3m55s ago)  72m
2022-04-30 04:12:18 file-level-Info triton.service.policy 0.0.4 is running dev mode with debug log level.
```

Figure 24-After Doing Modification On the Kong Gateway the gateway crashed everytime.

• Lessons learned and experience gained

Learned about the importance of having a gateway to the business product.In the triton project, since we are developing a Cloud security application having a gateway is much needed.But the issue was to do modification; we have to be expert in Kong. Otherwise, the gateway would malfunction.

2.2.4.Task 4 – Update the Policy Compliances-sample Data Feed v1.00.xlsx

This was a collaborative task done with one of the interns. In this task, we have to fill data into google sheets, which contain 11 tables. We have to investigate the Triton confluence pages where all the policies and all the data are stated. We have to consider three compliance and rule repositories. they are-

1. AWS CIS v1.2.0
2. AWS Best Practice FW
3. PCI DSS 3.2.1

According to those Compliances and rule repositories, We have filled the tables.

Name	Date modified	Type	Size
automatedremediationtemplate.csv	5/15/2022 12:54 PM	Microsoft Excel C...	1 KB
availablecloud.csv	5/16/2022 1:15 AM	Microsoft Excel C...	1 KB
cloudasset.csv	5/16/2022 8:12 AM	Microsoft Excel C...	4 KB
cloudAssetCategory.csv	5/12/2022 2:28 PM	Microsoft Excel C...	1 KB
guidedremediationtemplate.csv	5/16/2022 1:11 AM	Microsoft Excel C...	112 KB
policyframework.csv	5/16/2022 1:12 AM	Microsoft Excel C...	1 KB
policyrule.csv	5/15/2022 7:02 PM	Microsoft Excel C...	51 KB
policyruleavailablecloud .csv	5/16/2022 12:32 AM	Microsoft Excel C...	4 KB
policyrulecategory.csv	5/16/2022 1:14 AM	Microsoft Excel C...	2 KB
remediationtype.csv	4/29/2022 10:46 AM	Microsoft Excel C...	1 KB
riskcategory.csv	5/11/2022 8:19 PM	Microsoft Excel C...	1 KB

Figure 25- 11 tables and ther names

But at the beginning, all the data wasn't included in the confluence pages .it was affected to the development process. Because it will have null values in most of the cells, sometimes Golang seeder and seeder helper was modified and had to put extra effort into filling those null cells with dummy values. Because null cells could not be identified by the golang CSV reader.

Therefore to solve the issue, I had to put a string value “NULL” for every cell which are not having any value.

2.2.5. Task 5 – Make a Golang program to read CSV files and extract data into struts

- **Introduction to the Task**

This task was initiated to me because, at that time, we were manually creating this sdata.go file using the data we put into the policy compliances-sample data feed v1.0.0.xlsx sheet. It was a challenging task to create structures manually. Therefore a separate go program was built to automate this manual process.

- **Background Knowledge Gathering**

In this task, I have to learn about Golang CSV reading library. Because the whole task depending on reading a CSV file. Therefore I have studied more about accessing a CSV cell value using Golang. Because there were both integers and strings in CSV files, we should identify whether the value is Integer or a string while Copying a cell value into a struct. If a string, we should print double quotes while writing inside the structure.

- **Design of the component**

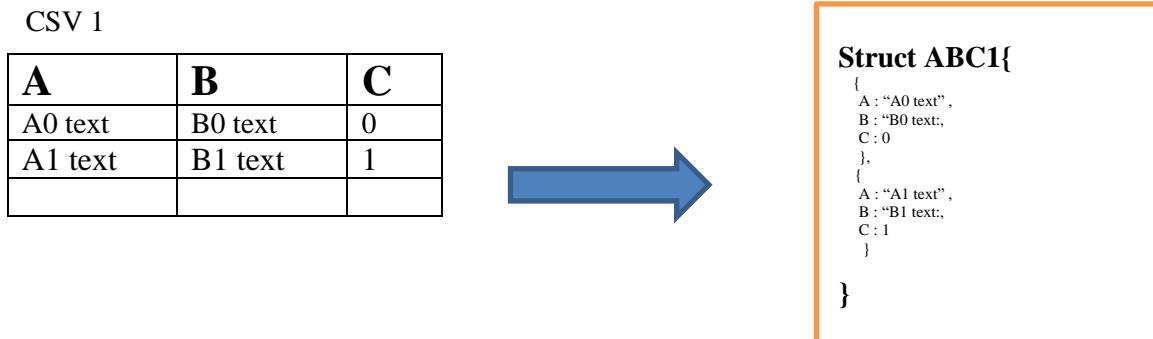


Figure 26-Seeder helper working concept

Likewise, by reading each row, in all 11 CSV files final sdata.go file was created.

- **Implementation and Testing of the Tool**

I have to check all the files has extracted into structs. It was a difficult task because After reading the CSV file data into the struct, the file line would get larger. Therefore, checking every line was not possible. For that, I have chosen several places and compared the written data with the Original CSV data.

```

file: sdata.go
content:
    package sdata

    type PolicyRuleAvailableCloudData struct {
        PolicyRuleID int
        CloudAssetID int
        PolicyRuleCategoryID int
    }

```

Figure 27-The generated sdata.go file

After creating this file, I added this file to the back-end and checked whether the data was seeding correctly without any errors.

- **Problems Faced during Implementation and Resolutions**

Several problems occurred because the CSV reading library has some limitations. Sometimes It wasn't working as expected. At one moment, we have put the values inside double quotations. But finally, doing Hard work, all the CSV data was written into the sdata.go file according to the correct order.

- **Lessons learned and experience gained**

This task taught me how to use the Golang CSV reader to extract data from a CSV file to a pure Golang file.

2.2.6. Task 6 – Modified the current E.R.D. of compliance rules repository

- **Introduction to the Task**

In the beginning, there was ER diagram for the Compliance and rules repository. But it was not optimized. We had to only modify the ER diagram according to the comment given by the principal architect.

- **Background Knowledge Gathering**

Before doing the task, I learned about the tables of the database and what they are for. Below are some findings I found from the confluence pages.

Table 6- Policy & compliance table details

Database table	Context
PolicyFramework	Keep all the supported Policy frameworks
Policy Rule Category	Keeps all policy/Rule Category
Policy Rule	Keeps the policy rules-related things
Risk category	Keeps the risk category of the policy rule
AvailableCloud	Keeps the available cloud for the relevant policy
PolicyRuleAvailableCloud	Mapping table of the policyRule and AvailableCloud
CloudAsset	Keeps all the Cloud Assets
CloudAssetCategory	Keeps the relevant category of the cloud asset
GuidedRemediationTemplate	-
AutomatedRemediationTemplate	-

Mainly the normalization techniques were used to optimize the relational database. On the other hand, the project's principal architect told us that every policy would have only guided remediationtemplateId and not need automatedremediationtemplateId. Therefore, we decided to remove the remediation type and AutomatedRemediationTemplate table from the current Er diagram

- **Design of the component**

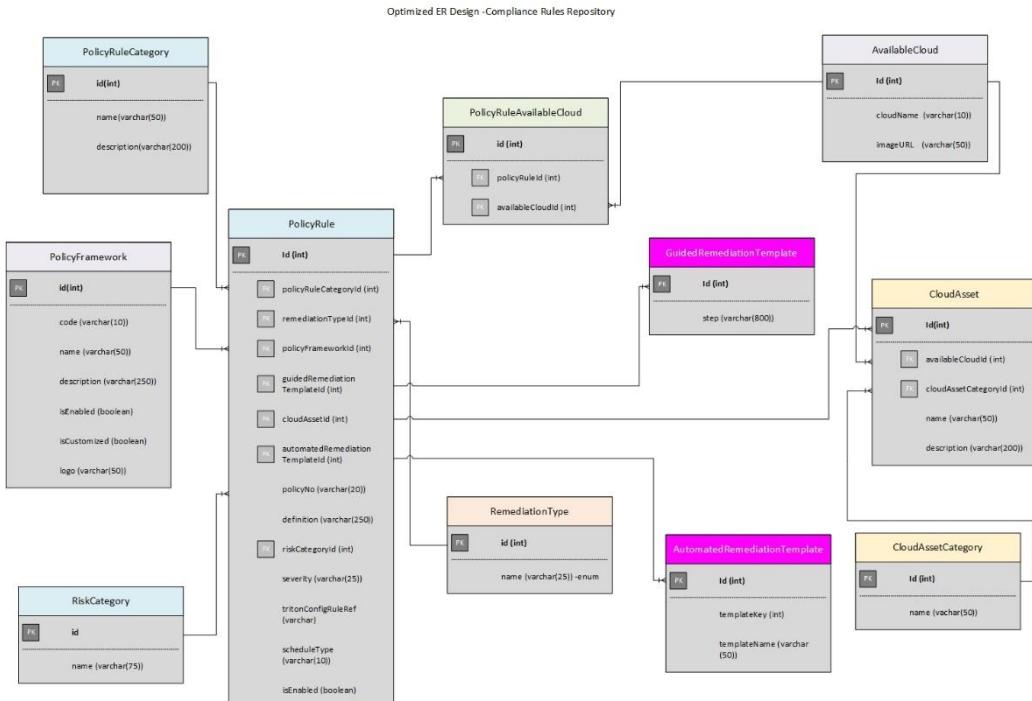


Figure 28-Modified and optimized ER diagram

- **Lessons learned and experience gained**

I have learned how to design relational databases and use normalization techniques to optimize them. On the other hand, I have learned how to draw ER diagrams using the Lucid chart and Visio tool.

2.2.7.Task 7–Update Golang data seeder according to the new changes

• Introduction to the Task

Another developer was initial build a data seeder using Golang. My task was to modify the seeder after all the database structure changes.

• Background Knowledge Gathering

At the moment, I was familiar with Golang, therefore, it was easy for me to read the codes. However, to change the database, I have to study Gorm. For that, I have to look at the Gorm official documentation.

• Implementation and Testing of the Tool

I didn't change the base model for the implementation; only what I changed was the tables field and their relations. According to ERD, the table columns and relations were changed successfully. In the project Gorm, auto migration technique was used.

Figure 29- dao.go file where the database Types were written

After doing the only way we could check the seeder function correctly was logs showing in the terminal while building the project.

Figure 30-Affter several modifications seeder sucessfully worked

Finally, to check the seeder function well, I had to connect the PostgreSQL database using k8s port forward method. To create a connection with it, I used the following command.

Kubectl port-forward service/triton/triton-db-postgresql 5432:5432

After connecting as figure 13, all the table's data should be there. Several times I had to modify until all tables seeded data in the correct format.

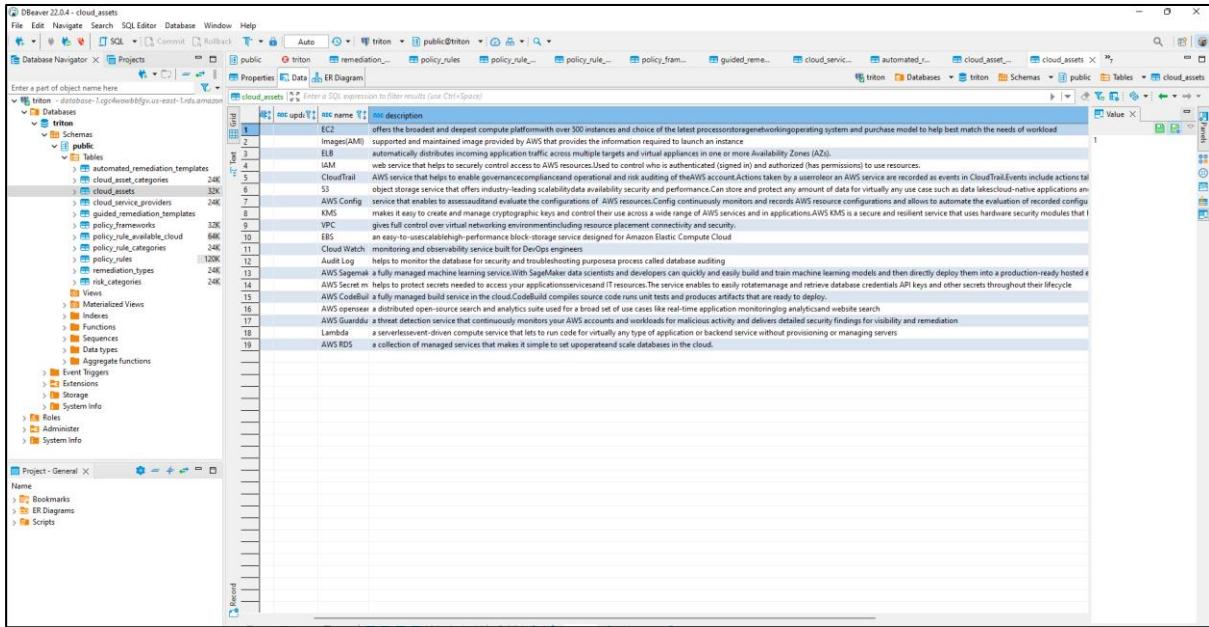


Figure 31-Connected to the Posgredql database image using K8s port forward method

Finally, we can generate the ER diagram usingdbeaver software. Using that, I have validated that the current database is as the ER Diagram drawn previously.

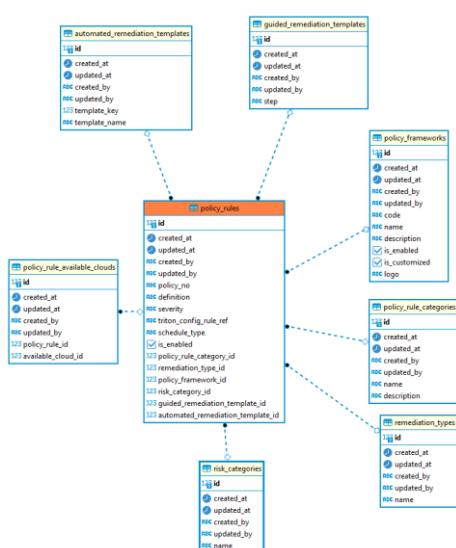


Figure 32-Policy_rules table and its relations

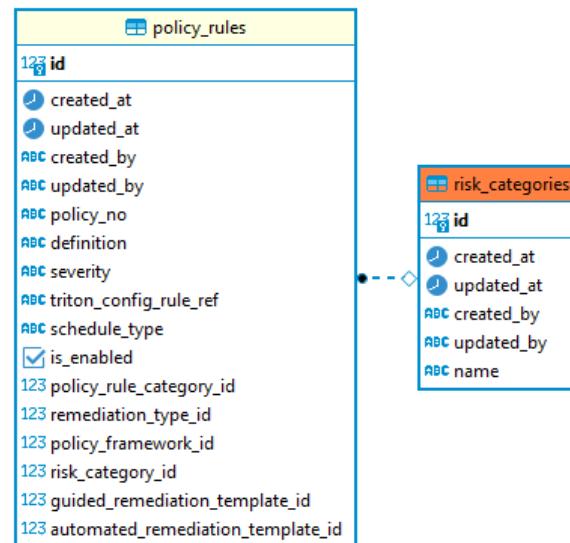


Figure 33-risk_categories table and its relations

• Problems Faced during Implementation and Resolutions

When I was doing the task several versions of the project has been released. Therefore some new code were merge to the main branch . Once we get a pull request from it all the newly added changes are merge into our local working branch. But while we building some of those new features need libraries to be install in the project building environment .If there wasn't the nessasry packages and libraries build would be unsuccessful at the middle.below was an error ocuured due to the unavailability of protobuf, a free and open-source cross-platform data format used to serialize structured data. Sometimes while building, the necessary files would download at the first ,but according to the quality of the internet connection, sometimes it fails to download them.

Figure 34-Error occurred due to the unavailability of some libraries installed in the working environment.

To escape the error, I have commented on some executing commands on the make file using protobuf packages.

```
File Edit Selection View Go Run Terminal Help services.mk - triton-microservice-backend [WSL: Ubuntu] - Visual Studio Code

SOURCE CON... 3 ✓ ... go sum M m services.mk M M versions.mk M M Makefile M M services.mk [Working Tree] M ... Message (Ctrl+Enter to commit ...)

Message (Ctrl+Enter to commit ...)
Staged Changes 8
Changes 5
M go.mod M
M gosum M
clean-policy-service:
rm -rf services/policy/.bin
build-policy-service-image: build-policy-service
docker build --no-cache -f services/policy/Dockerfile service
#####
## CLOUDSCANNER SERVICE ##
generate-cloudscanner-protobuf:
protoc --plugin=protoc-gen-go=$(GOPATH)/bin/protoc-gen-go \
--go_out=plugins=grpc \
build-cloudscanner-service: generate-cloudscanner-protobuf \
$(GO_BUILD_ENVIRONMENT) go build -o services/cloudscanner
clean-cloudscanner-service:
rm -rf services/cloudscanner/.bin
build-cloudscanner-service-image: build-cloudscanner-service
docker build --no-cache -f services/cloudscanner/Dockerfile \
#####
## CLOUDSCANNER SERVICE AWS ##
build-cloudscanner-aws-service:
$(GO_BUILD_ENVIRONMENT) go build -o services/cloudscanner
d/ ## POLICY SERVICE ##
build-policy-service:
$(GO_BUILD_ENVIRONMENT) go build -o services/policy/bin
clean-policy-service:
rm -rf services/policy/.bin
build-policy-service-image: build-policy-service
docker build --no-cache -f services/policy/Dockerfile service
#####
## CLOUDSCANNER SERVICE ##
generate-cloudscanner-protobuf:
protoc --plugin=protoc-gen-go=$(GOPATH)/bin/protoc-gen-go \
--go_out=plugins=grpc \
build-cloudscanner-service: generate-cloudscanner-protobuf \
$(GO_BUILD_ENVIRONMENT) go build -o services/cloudscanner
clean-cloudscanner-service:
rm -rf services/cloudscanner/.bin
build-cloudscanner-service-image: build-cloudscanner-service
docker build --no-cache -f services/cloudscanner/Dockerfile \
#####
## CLOUDSCANNER SERVICE AWS ##
build-cloudscanner-aws-service:
$(GO_BUILD_ENVIRONMENT) go build -o services/cloudscanner

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS
pasindu@SL-Puduweela:/mnt/c/Users/puduweela/appData/Local/Programs/Microsoft VS Code$
```

Figure 35-commented some line in the service.mk file to escape the error temporally

Another error was due to some incorrect relations the seeder terminates. Only the solution was I had to read the error, find the issue, and correct it.

```

pasindu@SL-PUDuwela: ~/triton-microservice-backend
Context: minikube
Cluster: minikube
User: minikube
K9s Rev: v0.25.18
K8s Rev: v1.23.3
CPU: n/a
MEM: n/a
Logs(triton-dev/policy-service-587cfccf89c-gkShc:policy-service)[1m]
Autoscroll:On FullScreen:Off Timestamps:Off Wrap:Off

2022-04-11 03:18:12 file-policy/main.go:40 level=info triton.service.policy 0.0.2 is running dev mode with debug log level.

[GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.
- using env: export GIN_MODE=release
- using code: gin.SetMode(gin.ReleaseMode)

2022/04/11 03:18:13 /home/pasindu/go/pkg/mod/gorm.io/driver/postgres/v1.3.1/migrator.go:118 ERROR: relation "policy_rules" does not exist (SQLSTATE 42P01)
[14.280ms] [rows:0] CREATE TABLE "cloud_assets" ("id" bigserial,"created_at" timestampz,"updated_at" timestampz,"created_by" varchar(50),"updated_by" varchar(50),
2022-04-11 03:18:13 file-policy/main.go:65 level=Fatal Unable to migrate the DB, ERROR: relation "policy_rules" does not exist (SQLSTATE 42P01)
Stream closed EOF for triton-dev/policy-service-587cfccf89c-gkShc (policy-service)

```

Figure 36-Error occurred due to Incorrect table relations

Sometimes some newly added column data size was not enough. Because the reason is in the tsv there was a huge paragraph. Therefore only the solution was to allocate the maximum size for varchar.

```

pasindu@SL-PUDuwela: ~
Context: minikube
Cluster: minikube
User: minikube
K9s Rev: v0.25.18
K8s Rev: v1.23.3
CPU: n/a
MEM: n/a
Logs(triton-dev/policy-service-7d99d89fb-5xfcvc:policy-service)[1m]
Autoscroll:On FullScreen:Off Timestamps:Off Wrap:Off

2022-04-27 02:34:05 file-policy/main.go:40 level=info triton.service.policy 0.0.3 is running dev mode with debug log level.

[GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.
- using env: export GIN_MODE=release
- using code: gin.SetMode(gin.ReleaseMode)

2022-04-27 02:34:06 file-dbutil/dbutil.go:39 level=info Seeding DB with default values...
2022-04-27 02:34:06 file-dbutil/dbutil.go:90 level=info Seeding Policy Framework...
2022-04-27 02:34:06 file-dbutil/dbutil.go:118 level=info No need to seed the Policy Framework, as table already contains Policy Framework
2022-04-27 02:34:06 file-dbutil/dbutil.go:119 level=info Seeding Policy Rule Category...
2022-04-27 02:34:06 file-dbutil/dbutil.go:131 level=info No need to seed the Policy Rule Category, as table already contains Rule Category
2022-04-27 02:34:06 file-dbutil/dbutil.go:133 level=info Seeding Risk Category...
2022-04-27 02:34:06 file-dbutil/dbutil.go:140 level=info No need to seed the Risk Category, as table already contains Risk Category
2022-04-27 02:34:06 file-dbutil/dbutil.go:198 level=info Seeding Available Cloud...
2022-04-27 02:34:06 file-dbutil/dbutil.go:205 level=info No need to seed the Available Cloud, as table already contains Available Cloud
2022-04-27 02:34:06 file-dbutil/dbutil.go:219 level=info Seeding Remediation Type...
2022-04-27 02:34:06 file-dbutil/dbutil.go:226 level=info No need to seed the Remediation Type, as table already contains Remediation Type
2022-04-27 02:34:06 file-dbutil/dbutil.go:230 level=info No need to seed the Cloud Asset Category...
2022-04-27 02:34:06 file-dbutil/dbutil.go:231 level=info No need to seed the Cloud Asset Category, as table already contains Cloud- Asset Category
2022-04-27 02:34:06 file-dbutil/dbutil.go:278 level=info Seeding Cloud Asset...
2022-04-27 02:34:06 file-dbutil/dbutil.go:289 level=info No need to seed the Cloud Asset, as table already contains Cloud Asset
2022-04-27 02:34:06 file-dbutil/dbutil.go:240 level=info Seeding Guided Remediation Template...

2022/04/27 02:34:06 /home/pasindu/triton-microservice-backend/services/policy/internal/pkg/dbutil.go:251 ERROR: value too long for type character varying(500)
[1.842ms] [rows:0] INSERT INTO "guided_remediation_templates" ("created_at","updated_at","created_by","step") VALUES ('2022-04-27 02:34:06.07','2022-04-27 02:34:06.07','pasindu','0')
2022-04-27 02:34:06 file-policy/main.go:26 level=error Internal Error: dbutil.go:251: value too long for type character varying(500) (SQLSTATE 22001)
Stream closed EOF for triton-dev/policy-service-7d99d89fb-5xfcvc (policy-service)

```

Figure 37-Error occurred due to insufficient variable size

• Lessons learned and experience gained

Learned about the purpose of a seeder and familiarized me with GORM. It was a new experience because I had the chance to work with relational databases.

2.2.8. Task 8 – Fill out the A.W.S. managed config rules repository

For this task, I have to create a google sheet and add some CLI command that matches the rules. This task was crucial since the whole rule engine rules were defined according to the findings that I recorded in the google sheet. In the beginning, I have familiar with A.W.S. very quickly. Because reading the documents doesn't sense anything about the rule. Therefore to understand the rule, some background knowledge should need. For that first, I have Created A AWS free tier account. Then followed some youtube tutorials to become familiar with I.A.M., S3, and other services.

s3-bucket-public-read-prohibited

	Rule	Command 1	Command 2
1	The Block Public Access setting restricts public policies or the bucket policy does not allow public read access	<code>aws s3api get-public-access-block --bucket my-bucket</code> <code>"BlockPublicPolicy": true</code>	<code>aws s3api get-bucket-acl --bucket config-bucket-423703969986</code> <code>"Permission": "WRITE"</code>
2	The Block Public Access setting restricts public ACLs or the bucket ACL does not allow public read access.	<code>aws s3api get-public-access-block --bucket my-bucket</code> <code>"BlockPublicAcls": true</code>	<code>aws s3api get-bucket-acl --bucket config-bucket-423703969986</code> <code>"Permission": "WRITE"</code>
3	If the Block Public Access setting does not restrict public policies, AWS Config evaluates whether the policy allows public read access. If the policy allows public read access, the rule is noncompliant.	<code>aws s3api get-public-access-block --bucket my-bucket</code> <code>"BlockPublicPolicy": false</code>	<code>aws s3api get-bucket-acl --bucket config-bucket-423703969986</code> <code>"Permission": "READ/FULL_CONTROL"</code>
4	If the Block Public Access setting does not restrict public bucket ACLs, AWS Config evaluates whether the bucket ACL allows public read access. If the bucket ACL allows public read access, the rule is noncompliant.	<code>aws s3api get-public-access-block --bucket my-bucket</code> <code>"BlockPublicAcls": false</code>	<code>aws s3api get-bucket-acl --bucket config-bucket-423703969986</code> <code>"Permission": "READ/FULL_CONTROL"</code>

Figure 38-Conditions and it relavant command for S3-bucket read prohibited

Following shows a table for one s3-related rule and how I showed the rule conditions, and at which time the rule will be compliant. Likewise, I have to cover 200+ rules.

Rule	Command 1	Command 2	Status1	Status2	Status3	Output	
1	true	true	true	true	true	Compliant	
	false	true	true				
	true	false	true	true	true		
	false	false	false				
2	true	true	true	true	true	noncompliant	
	false	true	true				
	true	false	true	false	false		
	false	false	false				
3	true	true	true	true	true	noncompliant	
	false	true	false				
	true	false	false	false	false		
	false	false	false				
4	true	true	true	true	true		
	false	true	false				
	true	false	false	false	false		
	false	false	false				

Figure 39-Truth tabel design for the relavant rule

In two weeks, I have covered 100+ rules related to EC2, I.A.M., S3 etc. All the findings were shared with the respective rule engine developer. After completing most of the rules, I was assigned to update rule.json file. The respective developer arranged some online sessions and told me what to do and how to do it. Below is a sample code given by the senior developer and I have to follow that pattern for the rule updates in the JSON file.

```

"s3-bucket-public-read-prohibited": {
    "level": "high",
    "combined_parameters": {
        "S3BucketPublicReadProhibited": {
            "and": [
                {
                    "or": [
                        {
                            "PublicAccessBlock": {
                                "Enabled": "is | true",
                                "BlockPublicPolicy": "is | true"
                            }
                        },
                        {
                            "Policy": {
                                "Statement": [
                                    {
                                        "Effect": "not matches | Allow",
                                        "Principal": "completely not matches | *,/*"
                                    }
                                ]
                            }
                        }
                    ]
                }
            ]
        }
    }
}

```

```

        "Action": "completely not matches | s3:*, s3:GetObject"
    }
]
}
}
],
{
"or": [
{
"PublicAccessBlock": {
"Enabled": "is | true",
"BlockPublicAcls": "is | true"
}
},
{
"A.C.L.": {
"Grants": [
{
"Permission": "completely not matches | READ,FULL_CONTROL"
}
]
}
}
]
}
]}

```

While Filling out the google sheet, I have faced some difficulties because some rules dosen't have definite one command. So we have read the rule and identified the conditions and create a truth table for it. Because some rules have more than one conditions and check all the conditions to identify whether the rule is a complaint or not.

2.2.9.Task 9 – Make an ETL to update data from the tsv format file to Cloud Database

- **Introduction to the Task**

In the project, after deploying the application on AWS Elastic Kubernetes service, if the client needed to update the data-related policies from the database side, there were no possible methods to do it while the system was up and running. Therefore I have assigned a task to create an ETL to read files from the S3 bucket and load them into the RDS PostgreSQL database.

- **Background Knowledge Gathering**

In this task, I have to quickly become familiar with AWS services though I was not using AWS for a long time. From my personal account, I have to learn to create the lambda functions and the PostgreSQL database. Python is also new to me; therefore, I had to follow some tutorials on python basics.

- **Design of the component**

To create A lambda function, We have to create an IAM role with access to the AWS lambda feature, RDS full access, cloud formation services, and s3 all access. After creating the role, we have to use that role in our lambda function to access each service at necessary times. We need to configure the API gateway if we want to trigger the lambda function using a specific endpoint URL.

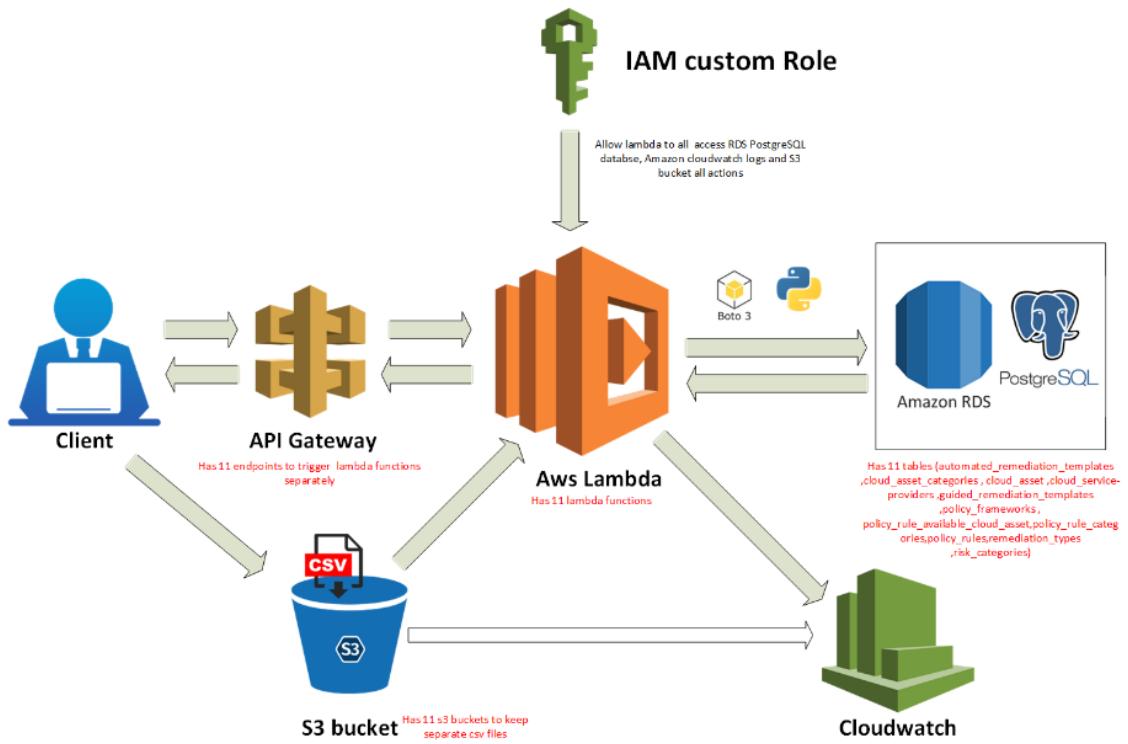


Figure 40-Simple diagram that shows the working flow of the ETL

Below are some pre-works that I have to show to the project's Principal architect to show my understanding about the given task.

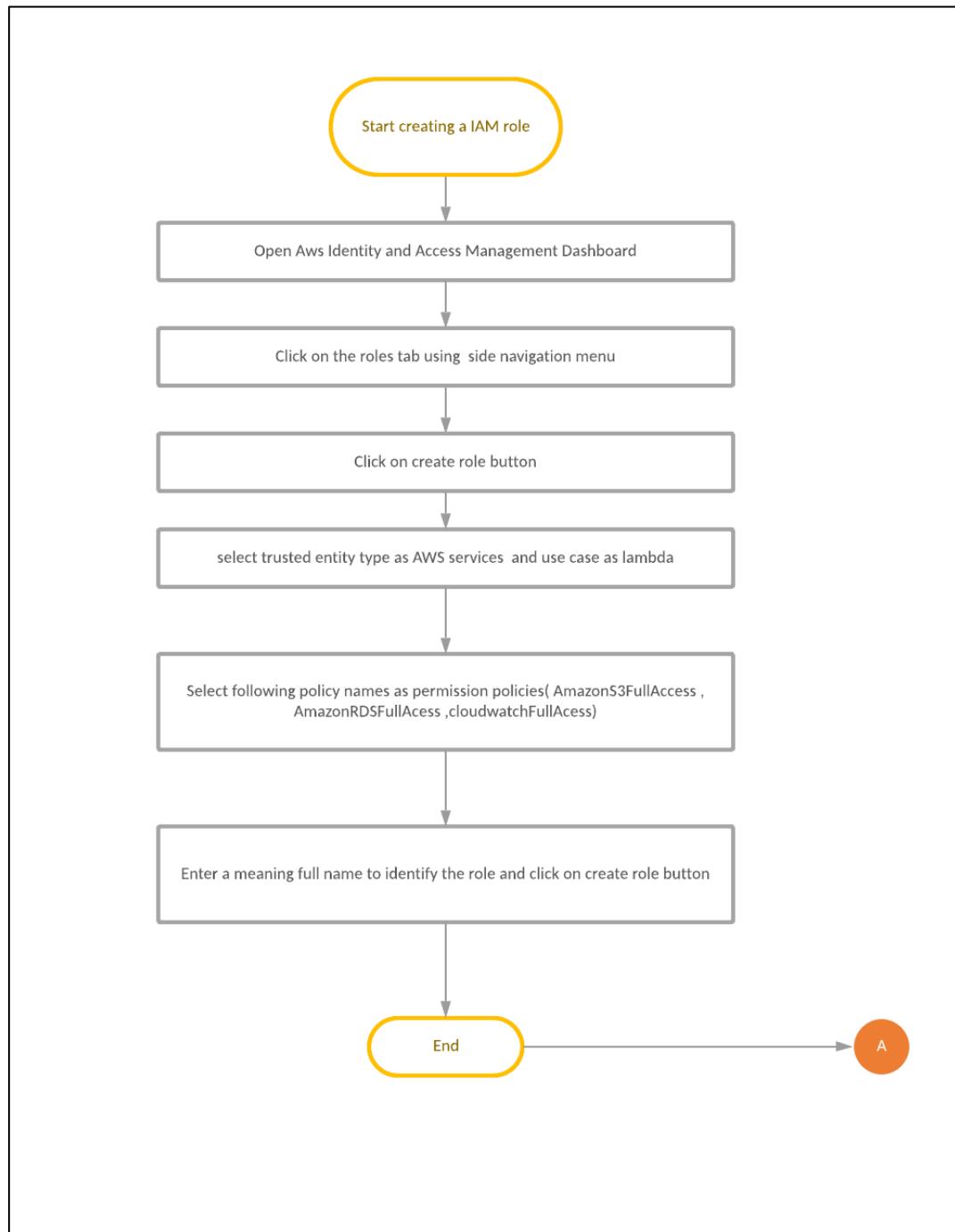


Figure 41-Work flow digram On Craeting IAM Custom Role

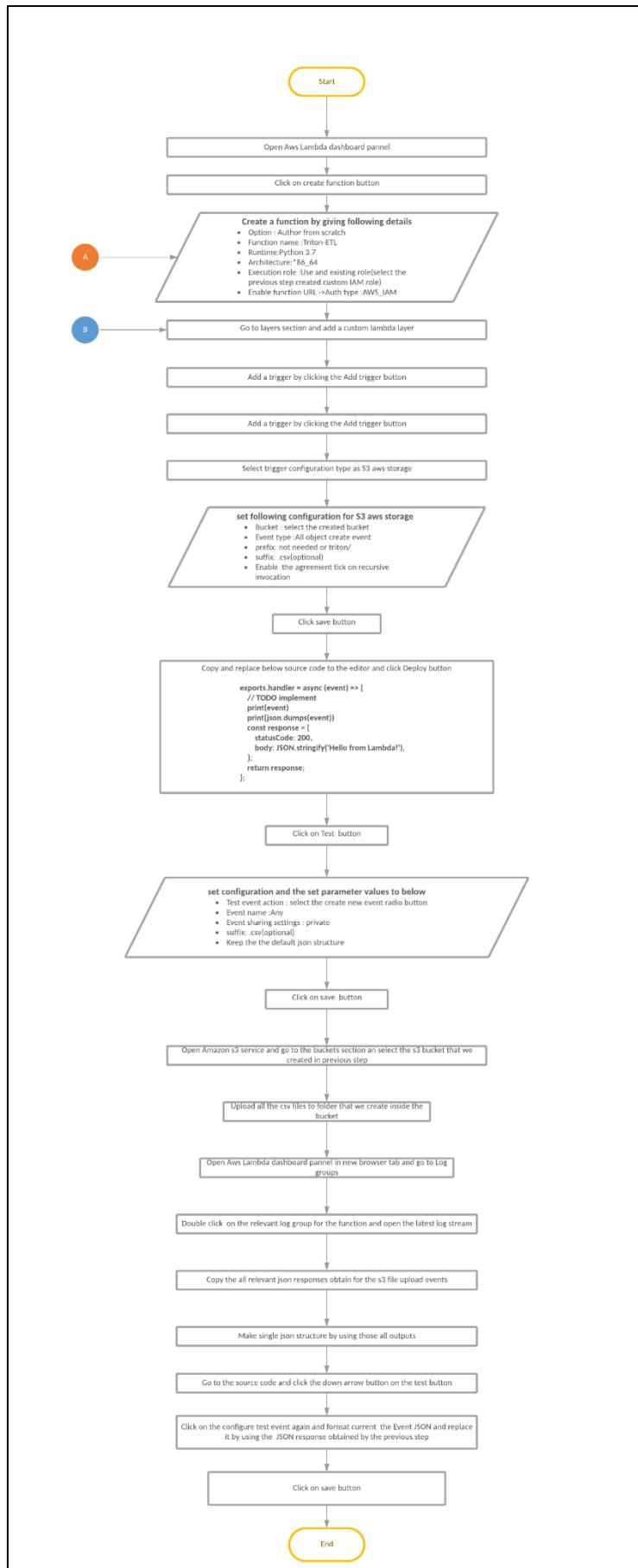


Figure 42-Flow diagram to Get the event Json response using CloudwatchLogs(Not necessary we can obtain those data using AWS dashboard)

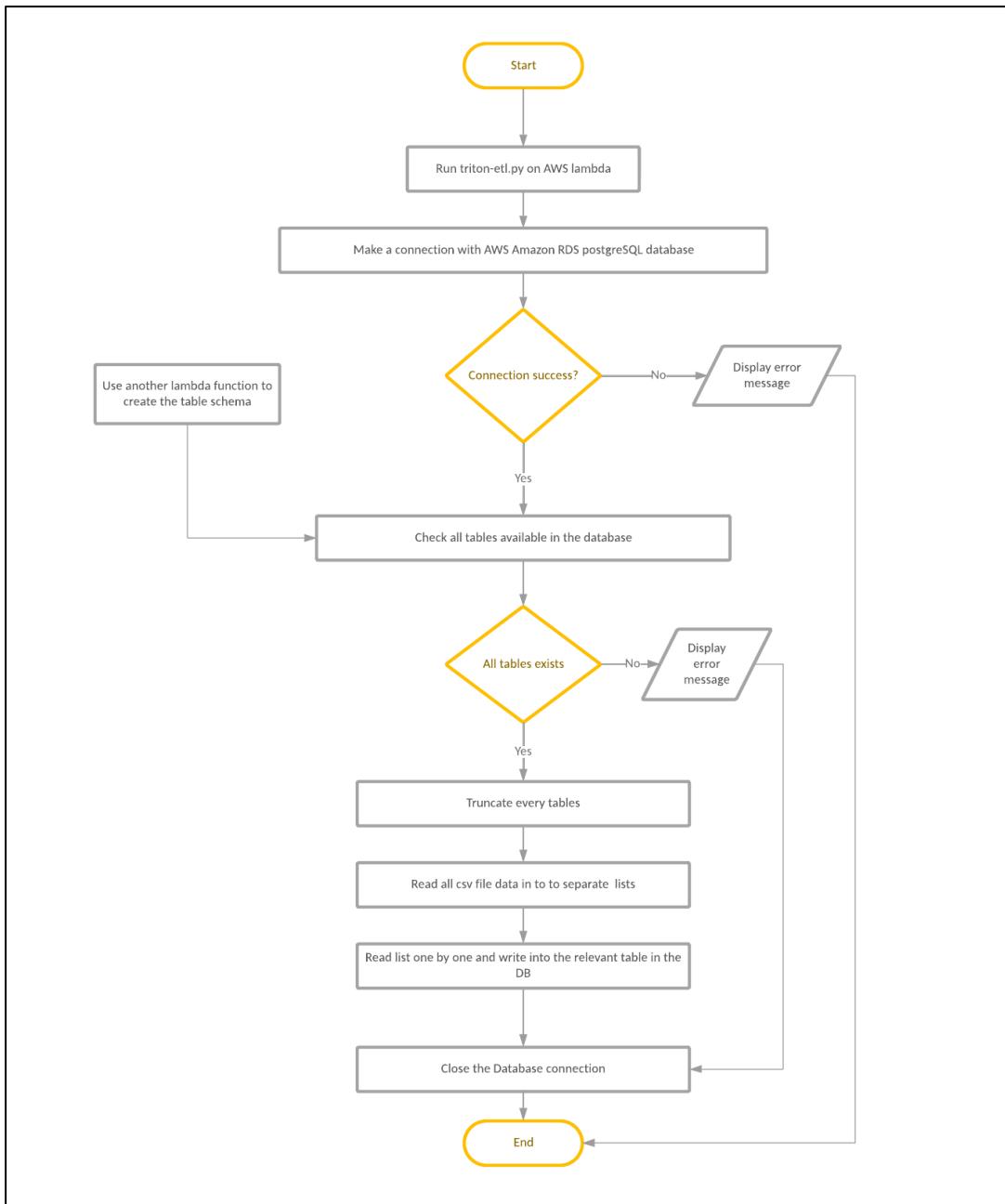


Figure 43- Triton ETL WorkFlow Diagram

• Implementation and Testing of the Tool

For the implementation, I had to use which was an AWS SDK for Python (Boto3) to create, configure, and manage AWS services, such as Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Simple Storage Service (Amazon S3). The SDK provides an object-oriented API as well as low-level access to AWS services. All the coding was done on the AWS given IDE because we connect with all services by using the graphical interface.

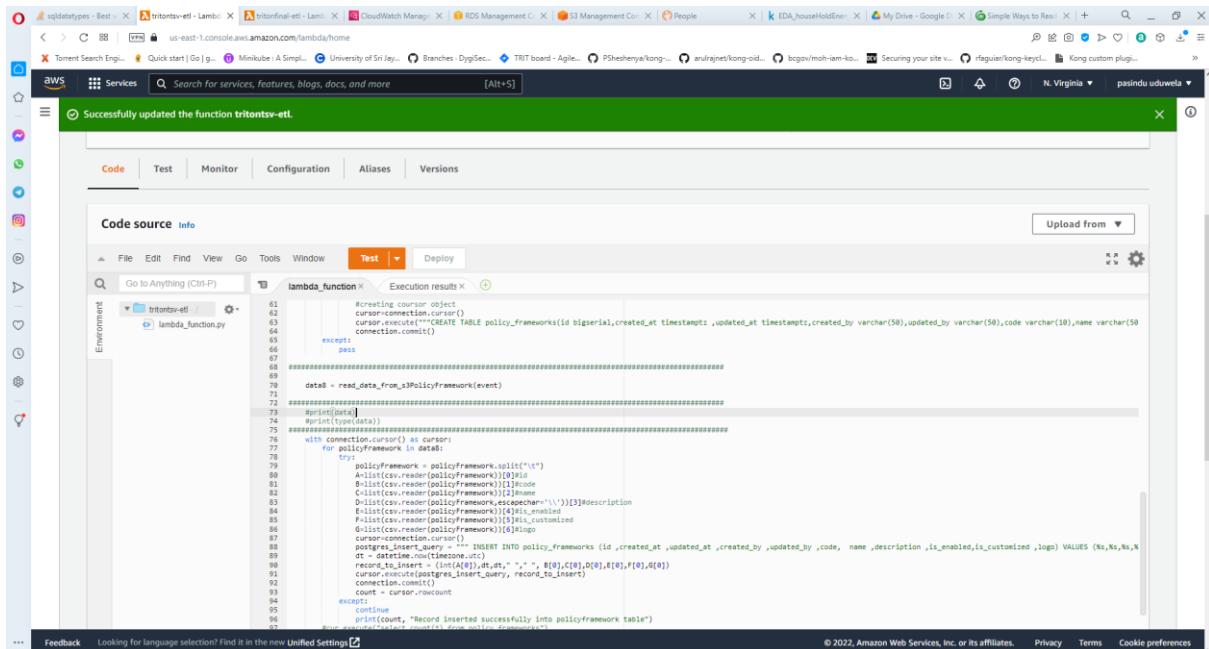


Figure 44-Online IDE which AWS provide us to test our lambda functions

• Problems Faced during Implementation and Resolutions

The main issue was that I had to test the ETL on my personal AWS account. Since AWS metrics are not showing correct billing, I had to be very careful on using services. For as a solution, I have set billing alarms. Another issue I had occurred was a time-out error while testing the ETL. For that, I have found a solution from StackOverflow and increased the memory size and the execution duration allocated to the ETL. While increasing those parameters, AWS will automatically monitor your AWS account and warn of fair usage.

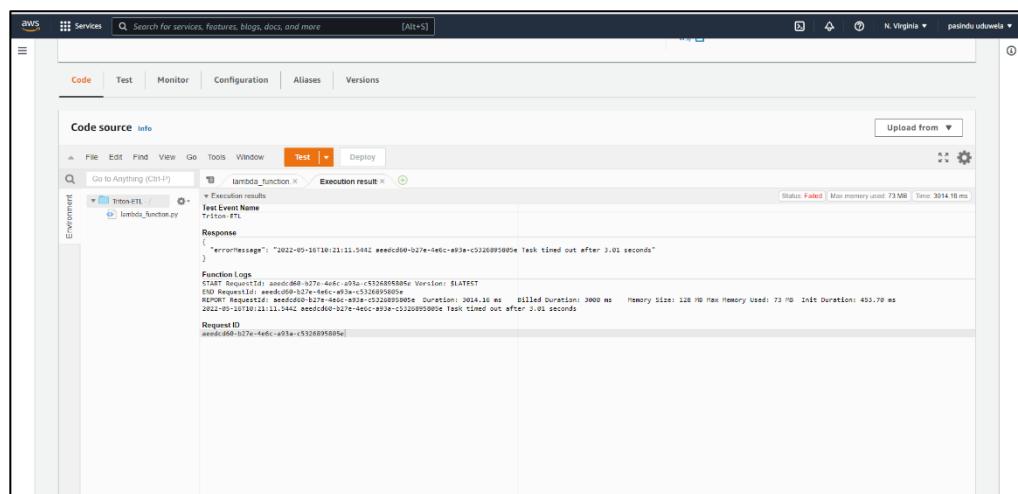


Figure 45-Time Out Error occurred while developing the ETL

Another issue was I was not able to connect to the PostgreSQL database using the port number and the source IP. I have to read the AWS documentation and find a solution. Finally, I have

found that we can expose our database to the outside world by adding an inbound rule to the relevant database-connected VPC security group. Then AWS allows us to access the database from a local machine. Another issue was At the beginning, I was planning to use CSV files by directly generating from the Compliances-sample Data Feed v1.00.xlsx google sheet.But while working, I have identified CSV reading is not possible because in some tables, that are paragraphs that contain commas inside them. As a result csv reader identifies those commas as a line break and divides them into columns. Then the final output got messy and sometimes, even though I used the escape char mechanism, the commas inside the paragraphs would not be shown correctly.Therefore by the comments given by the tech lead, I have to use tab-separated files instead of using comma-separated files. It worked for the ETL.

The screenshot shows a database interface with a table. The first column has a header 'Practices'. The second column contains several rows of text. One row, which starts with 'The CIS Security Benchmarks program provides well-defined' and ends with 'some security best practices to help protect your AWS resources.', has a red box drawn around the end of the first sentence and the beginning of the second sentence, highlighting a missing character (likely a newline or a space). The table has a header row and several data rows below it. On the right side of the interface, there is a toolbar with various icons and a vertical panel labeled 'Panels' containing several small icons.

Figure 46-Due to commas inside paragraphs some characters were missing after loading the data to database.

- Lessons learned and experience gained**

I have used IAM, S3, and Cloudwatch log services for the ETL. Therefore I have to learn those deeply. Next, I learned about the AWS lambda service. In lambda service, I coded the ETL using python and used boto3 also. Then learned about how to add lambda layers to lambda functions. On the other hand, I had the chance to polish up my SQL knowledge also .because the whole database structure was created using one SQL script.

2.2.10. Task 10 – Make a YAML to automate the ETL deploying process

- Introduction to the Task**

Here the task was to automate the ETL deployment process; for that, we had to use the AWS cloud formation service. In this project, I created the YAML. However, we can use either YAML or JSON to automate the process. For this task, I have to use the AWS cloud formation service. Therefore I have to follow some youtube videos to familiarize myself with the service. Due to legal issues, the Full code isn't copied to the report. Therefore I have shown a part of YAML.

```

File Edit Selection View Go Run Terminal Help
triton deploy.yaml Visual Studio Code
L > Triton-works > LUMIDA > s3-bucket-public-read-prohibited.go | / triton-deploy.yaml | x
1 #!/bin/bash
2 #!/bin/bash
3 #!/bin/bash
4 #!/bin/bash
5 #!/bin/bash
6 #!/bin/bash
7 #!/bin/bash
8 #!/bin/bash
9 #!/bin/bash
10 #!/bin/bash
11 #!/bin/bash
12 #!/bin/bash
13 #!/bin/bash
14 #!/bin/bash
15 #!/bin/bash
16 #!/bin/bash
17 #!/bin/bash
18 #!/bin/bash
19 #!/bin/bash
20 #!/bin/bash
21 #!/bin/bash
22 #!/bin/bash
23 #!/bin/bash
24 #!/bin/bash
25 #!/bin/bash
26 #!/bin/bash
27 #!/bin/bash
28 #!/bin/bash
29 #!/bin/bash
30 #!/bin/bash
31 #!/bin/bash
32 #!/bin/bash
33 #!/bin/bash
34 #!/bin/bash
35 #!/bin/bash
36 #!/bin/bash
37 #!/bin/bash
38 #!/bin/bash
39 #!/bin/bash
40 #!/bin/bash
41 #!/bin/bash
42 #!/bin/bash
43 #!/bin/bash
44 #!/bin/bash
45 #!/bin/bash
46 #!/bin/bash
47 #!/bin/bash
48 #!/bin/bash
49 #!/bin/bash
50 #!/bin/bash
51 #!/bin/bash
52 #!/bin/bash
53 #!/bin/bash
54 #!/bin/bash
55 #!/bin/bash
56 #!/bin/bash
57 #!/bin/bash
58 #!/bin/bash
59 #!/bin/bash
60 #!/bin/bash
61 #!/bin/bash
62 #!/bin/bash
63 #!/bin/bash
64 #!/bin/bash
65 #!/bin/bash
66 #!/bin/bash
67 #!/bin/bash
68 #!/bin/bash
69 #!/bin/bash
70 #!/bin/bash
71 #!/bin/bash
72 #!/bin/bash
73 #!/bin/bash
74 #!/bin/bash
75 #!/bin/bash
76 #!/bin/bash
77 #!/bin/bash
78 #!/bin/bash
79 #!/bin/bash
80 #!/bin/bash
81 #!/bin/bash
82 #!/bin/bash
83 #!/bin/bash
84 #!/bin/bash
85 #!/bin/bash
86 #!/bin/bash
87 #!/bin/bash
88 #!/bin/bash
89 #!/bin/bash
90 #!/bin/bash
91 #!/bin/bash
92 #!/bin/bash
93 #!/bin/bash
94 #!/bin/bash
95 #!/bin/bash
96 #!/bin/bash
97 #!/bin/bash
98 #!/bin/bash
99 #!/bin/bash
100 #!/bin/bash
101 #!/bin/bash
102 #!/bin/bash
103 #!/bin/bash
104 #!/bin/bash
105 #!/bin/bash
106 #!/bin/bash
107 #!/bin/bash
108 #!/bin/bash
109 #!/bin/bash
110 #!/bin/bash
111 #!/bin/bash
112 #!/bin/bash
113 #!/bin/bash
114 #!/bin/bash
115 #!/bin/bash
116 #!/bin/bash
117 #!/bin/bash
118 #!/bin/bash
119 #!/bin/bash
120 #!/bin/bash
121 #!/bin/bash
122 #!/bin/bash
123 #!/bin/bash
124 #!/bin/bash
125 #!/bin/bash
126 #!/bin/bash
127 #!/bin/bash
128 #!/bin/bash
129 #!/bin/bash
130 #!/bin/bash
131 #!/bin/bash
132 #!/bin/bash
133 #!/bin/bash
134 #!/bin/bash
135 #!/bin/bash
136 #!/bin/bash
137 #!/bin/bash
138 #!/bin/bash
139 #!/bin/bash
140 #!/bin/bash
141 #!/bin/bash
142 #!/bin/bash
143 #!/bin/bash
144 #!/bin/bash
145 #!/bin/bash
146 #!/bin/bash
147 #!/bin/bash
148 #!/bin/bash
149 #!/bin/bash
150 #!/bin/bash
151 #!/bin/bash
152 #!/bin/bash
153 #!/bin/bash
154 #!/bin/bash
155 #!/bin/bash
156 #!/bin/bash
157 #!/bin/bash
158 #!/bin/bash
159 #!/bin/bash
160 #!/bin/bash
161 #!/bin/bash
162 #!/bin/bash
163 #!/bin/bash
164 #!/bin/bash
165 #!/bin/bash
166 #!/bin/bash
167 #!/bin/bash
168 #!/bin/bash
169 #!/bin/bash
170 #!/bin/bash
171 #!/bin/bash
172 #!/bin/bash
173 #!/bin/bash
174 #!/bin/bash
175 #!/bin/bash
176 #!/bin/bash
177 #!/bin/bash
178 #!/bin/bash
179 #!/bin/bash
180 #!/bin/bash
181 #!/bin/bash
182 #!/bin/bash
183 #!/bin/bash
184 #!/bin/bash
185 #!/bin/bash
186 #!/bin/bash
187 #!/bin/bash
188 #!/bin/bash
189 #!/bin/bash
190 #!/bin/bash
191 #!/bin/bash
192 #!/bin/bash
193 #!/bin/bash
194 #!/bin/bash
195 #!/bin/bash
196 #!/bin/bash
197 #!/bin/bash
198 #!/bin/bash
199 #!/bin/bash
200 #!/bin/bash
201 #!/bin/bash
202 #!/bin/bash
203 #!/bin/bash
204 #!/bin/bash
205 #!/bin/bash
206 #!/bin/bash
207 #!/bin/bash
208 #!/bin/bash
209 #!/bin/bash
210 #!/bin/bash
211 #!/bin/bash
212 #!/bin/bash
213 #!/bin/bash
214 #!/bin/bash
215 #!/bin/bash
216 #!/bin/bash
217 #!/bin/bash
218 #!/bin/bash
219 #!/bin/bash
220 #!/bin/bash
221 #!/bin/bash
222 #!/bin/bash
223 #!/bin/bash
224 #!/bin/bash
225 #!/bin/bash
226 #!/bin/bash
227 #!/bin/bash
228 #!/bin/bash
229 #!/bin/bash
230 #!/bin/bash
231 #!/bin/bash
232 #!/bin/bash
233 #!/bin/bash
234 #!/bin/bash
235 #!/bin/bash
236 #!/bin/bash
237 #!/bin/bash
238 #!/bin/bash
239 #!/bin/bash
240 #!/bin/bash
241 #!/bin/bash
242 #!/bin/bash
243 #!/bin/bash
244 #!/bin/bash
245 #!/bin/bash
246 #!/bin/bash
247 #!/bin/bash
248 #!/bin/bash
249 #!/bin/bash
250 #!/bin/bash
251 #!/bin/bash
252 #!/bin/bash
253 #!/bin/bash
254 #!/bin/bash
255 #!/bin/bash
256 #!/bin/bash
257 #!/bin/bash
258 #!/bin/bash
259 #!/bin/bash
260 #!/bin/bash
261 #!/bin/bash
262 #!/bin/bash
263 #!/bin/bash
264 #!/bin/bash
265 #!/bin/bash
266 #!/bin/bash
267 #!/bin/bash
268 #!/bin/bash
269 #!/bin/bash
270 #!/bin/bash
271 #!/bin/bash
272 #!/bin/bash
273 #!/bin/bash
274 #!/bin/bash
275 #!/bin/bash
276 #!/bin/bash
277 #!/bin/bash
278 #!/bin/bash
279 #!/bin/bash
280 #!/bin/bash

```

Figure 47-YAML file of the ETL

- **Implementation and Testing of the Tool**

After creating the ETL again, I had to check that the ETL was working correctly. I have used dbeaver software and connected to the AWS PostgreSQL relational database. Finally, all the data in the tsv files were extracted into the database relevant tables without any change.

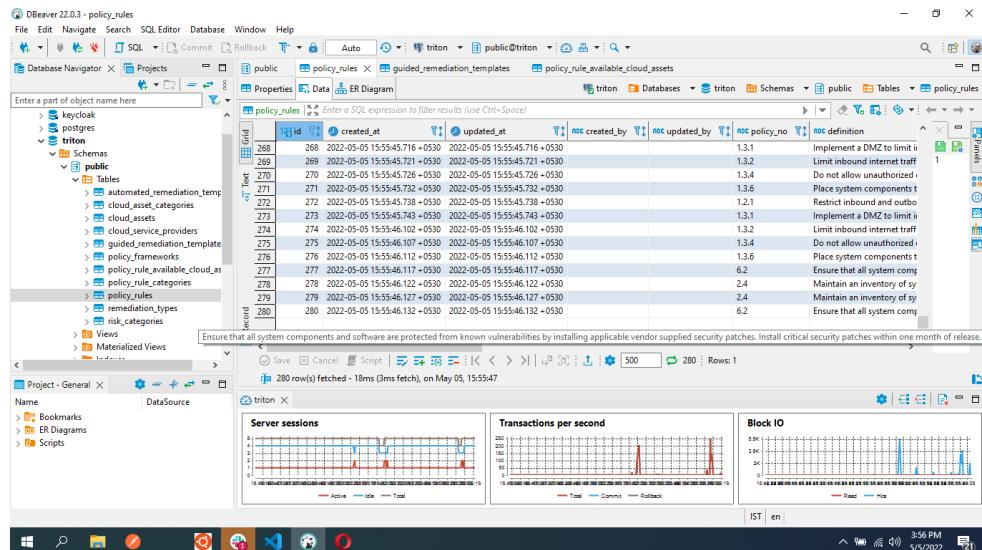


Figure 48-After connecting the database to dbeaver

- **Problems Faced during Implementation and Resolutions**

The main issue was I had to set a billing alert to the cloud formation service since I was using the free tier; sometimes, if it exceeds the quota, AWS will charge higher rates. But unfortunately, I have charged 4.66 USD from me at the end of the project. The reason was the relational database instance wasn't stopped and that service runs continuously. Since the company doesn't give us an office AWS account, It cost me around 4000 rupees from my

debit card. Finally, all the payments have been settled and neutralised the AWS until another use.

7/3/22, 2:13 PM	Billing Management Console		
 AWS bill summary <small>Info</small>			
Total charges and payment information			
Account ID 001728929434	Billing period <small>Info</small> June 1 - June 30, 2022		
Service provider Amazon Web Services, Inc.	Invoice currency USD - US dollars	Total in invoice currency USD 4.66	Total in USD USD 4.66
Grand total:		USD 4.66	

Figure 49-AWS Monthly bill Summary

7/3/22, 2:13 PM	Billing Management Console
Amazon Web Services, Inc. charges by service <small>Info</small>	
Total active services 6	
Service name Region	Amount in USD
Relational Database Service	USD 4.66
CloudWatch	USD 0.00
Data Transfer	USD 0.00
Elastic Compute Cloud	USD 0.00
Lambda	USD 0.00
Simple Storage Service	USD 0.00
Total tax	USD 0.00

Figure 50-AWS Services used and their charges

2.3. Short overview of the Second Project (ATS - Application Tracking System)

As one of Mitra Innovation's main departments, the Human Resource Department is critical in recruiting employees for available positions in the organization. Currently, the Human Resource Department's staff manually maintains the Records of Recruitment with the

association of the PeopleHR application. Since Recruitments take place often, it becomes a tough job for the officials to maintain a higher level of accuracy along with good efficiency and confidentiality in updating them into the system. Also, there are high chances of encountering data losses and erroneous entries since they are done manually. To address the above-mentioned drawbacks and difficulties in the existing system, it is suggested to develop a web application that replaces the current manual data capturing and visualization of the recruitment process. Department expects to maintain the recruitment process by overcoming overheads in the existing manual system and visualize the statistics associated with recruitments over months and weeks.

In this Internal project, we have the freedom to design the system and show the work to the software architect. On the other hand, some udemy free tutorial links were shared with us to familiarize us with Springboot and angular. The software architect shared his experience while planning and designing the system. It helps a lot to speed up the designing process.

2.4. Tasks Assigned in the Second project

2.4.1.Task 1 – Make an ER diagram and a scheduler mechanism for the database

Initially, the plan was to create API endpoints using peopleHR and retrieve the data from PeopleHR Server in real-time. Nevertheless, that method was not practical for large amounts of data. Hence we expected to use the spring Jackson method for that. However, while gathering requirements and observing the peopleHR platform, we have seen that the API response time is too much larger, and the object size is more significant for the massive amount of data. Therefore we decided to create an intermediate relational database for the project. For that, I have observed the entities using the project and the sandbox environment and checked the attributes that we could expose from the peopleHR platform. Though we are leaving after six months, we were not given full access to the system except for two developers on the team.

The main challenge was to identify the relations with each table. For that, we have checked similar projects and marked the relations according to them. Similarly, we have identified current tables data one by one using the filter options from the peopleHR system. Then we recorded those data in a google sheet with the relevant foreign keys and mappings. It was a hard task since mapping all the data was not an easy task .therefore, we had to double-check the data that we collected to the google sheet was similar to the PeopleHR current database data.

We faced some difficulties while collecting data because the current peopleHR system has some issues in the primary key field. Therefore the same applicant might be entered into the system with two different applicant codes. The other issue was Mitra innovation Previous HR crew handled the system differently. So sometimes, they create the vacancy, update the applicant's details for the created vacancy, and select that vacancy at the same time. So it confused us because the system logs recorded for them were the same date. To Solve that issue, we have arranged some meetings with the HR team and discussed those issues. Then we

identified those issues that happened for the previous HR crew in the company before 2019, and now they manage the system very carefully. They told us now they created the vacancy first, and then once the applicant's cv is received by the company, they will upload the cv to google drive and enter the details into the system. so they keep track of all applicant's details now. Below is one table how it shows after collecting the data using the current peopleHR system. Likewise, all the data recorded in the google sheet were the same in the PeopleHR system.

id	stage_name
1	1st Interview
2	1st Interview Accepted/ Awaiting 1st Interview
3	1st Interview Scheduled
4	2nd Interview
5	2nd Interview Scheduled
6	Accepted
7	Aptitude Test
8	Awaiting Approval
9	Awaiting Case Study answers
10	Case Study
11	Client Interview
12	Declined Offer
13	Final Interview
14	Interview Attended decision pending
15	Invite to 1st Interview
16	Job Offer Declined
17	Job Offered

Figure 52-The part of data capture to the pipeline stages table

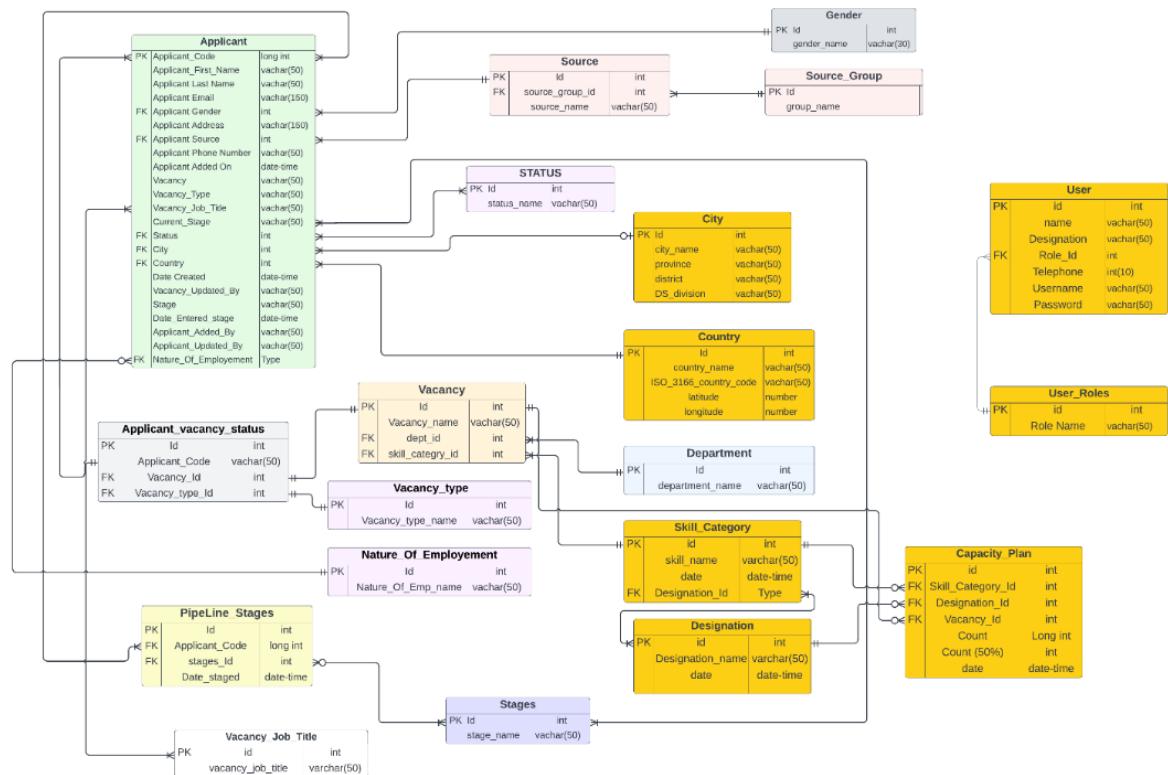


Figure 51-The ER diagram I designed for the Applicant tracking project

After completing the ERD, I tried to design the heart of this system. For that project's software architect gave us guidance. The scheduler's main purpose was to update the database in real-time. So I have researched it to check the feasibility of developing a scheduler for our system. So finally, I propose a workflow for the scheduler. The intermediate database is updated according to a sequence. On the other hand, I gave the manual scheduler invoking feature using a clickable button from the web interface. Because the scheduler invokes frequency by default would be four times per hour. If someone added the data from the peopleHR end, it would not be updated suddenly on our proposed intermediated database. Therefore if someone needs to generate reports 100% accurately, he/she can invoke the scheduler before generating the reports.

I have recorded some key points that we should consider while implementing the scheduler. They are-

- The scheduler should invoke four times per hour. (We decided to use the thread concept in the scheduler if the time was enough to develop a basic scheduler.)
- We decided to run the scheduler as one part of the system because we decided to build the scheduler in one container.
- At the initial build, we decided to use the Use Spring batch to read the CSV files for some tables, and the API endpoint obtained other tables' data. After that, All the tables will be updated using a specific API endpoint.

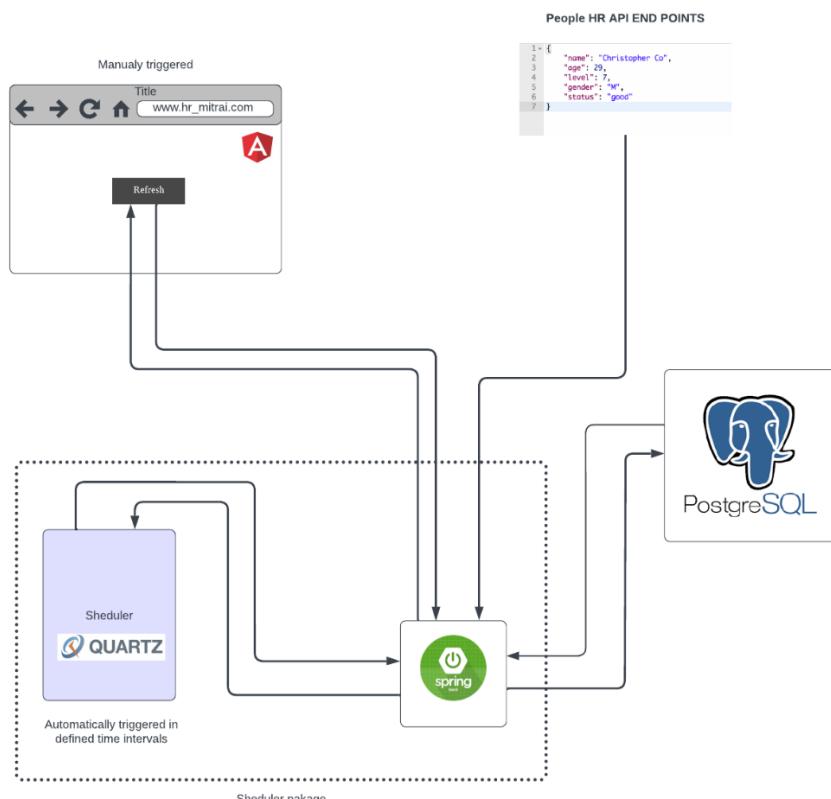


Figure 53-Work flow diagram of the scheduler

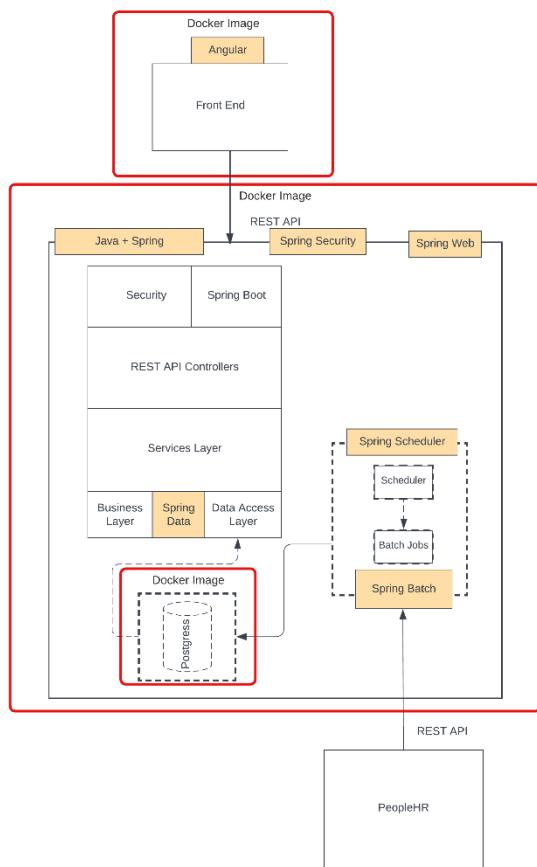


Figure 54-ATS Deployment Diagram

Figure 55-APIS created from PeopleHR end to get data to our database.

2.4.2. Task 2 – propose some wireframes to the team

Below are some wireframes I have designed using Figma Online UX/UI design tool.

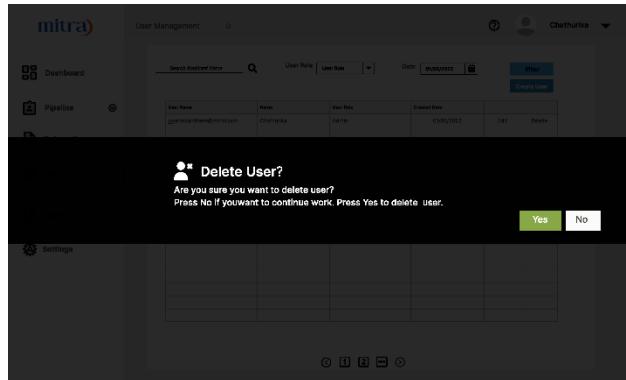


Figure 56-Delete user

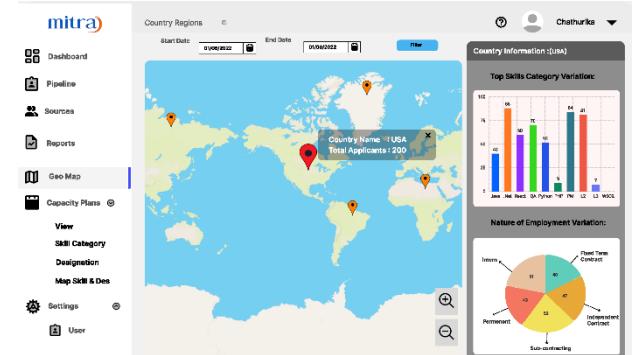


Figure 57-Countrywise skill category variation

This wireframe shows a form titled 'Manage Capacity Plans' under the 'Capacity Plans' section. It includes dropdown menus for 'Skill Category' and 'Designation', and input fields for 'Vacancy', 'Count', and 'Count(%)'. A 'Create' button is at the bottom.

Figure 58-Skill And Designation Mapping

This wireframe shows an edit view of the 'Manage Capacity Plans' form. It displays existing values for 'Skill Category: JMT', 'Designation: SE', 'Vacancy: Just SE', 'Count: 40', and 'Count(%)'. An 'Edit' button is visible at the bottom.

Figure 59- Edit Skill and designation Mapping

This wireframe shows a form titled 'Manage Designation' under the 'Designation' section. It has a dropdown for 'Skill Category' and input fields for 'Start Date' and 'End Date'. A large input field labeled 'Designation' is centered, with an 'Add' button below it.

Figure 60-Add Designation

This wireframe shows an edit view of the 'Manage Designation' form. It displays the 'Designation: SE' and an 'Edit' button at the bottom.

Figure 61-Edit Designation

2.4.3.Task 3 – Supported creating a 3P estimation sheet

This is a collaborative task assigned to all developers in which we have to fill the Mitra 3-point estimation sheet. The primary purpose is to get a rough idea about the project duration. This helps to draw the timeline using Atlassian project management tools.

The screenshot shows the 'Mitra Estimation Template.xlsx' spreadsheet in Excel. The main sheet is titled 'Detailed Estimation'. It has several sections:

- Project Information:** Includes fields for Client Name, HR Department, Purpose, Authorizer, Requestor, and Date Required By.
- Work Estimation Details:** A table with columns for Activity/Task, Activity/Task Breakdown, Task Type, and various estimation metrics (Optimistic, Most Likely, Pessimistic, Confidence, etc.) for 75%, 85%, 95%, and 99.5% completion levels. It also includes a column for 'Supporting Document Links'.
- Detailed Estimation:** A large table listing specific tasks with their estimated effort days, confidence levels, and links to external resources like Figma and YouTube.

			D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Mitra Innomation		Client Name: HR Department														Estimate Valid For: days
2	Description: Automate the data deriving process for the HR.		Purpose: HR Automation Project Time estimation														
3	Estimator:		Authorizer: (Enter Name, Title, Phone and Email address)														
4	Date Required By:		Requestor:														
5	Completion Date: 26/7/2022		Requested Date:														
6	Work Estimation Details																
7	#	Activity/Task	Activity/Task Breakdown	Task Type	Optimistic	Most Likely	Pessimistic	Confidence	75%	85%	95%	99.5%					
8	1	Login	Login Screen (frontend) login screen	Development	1.5	4.0	6.0	0.8	3.9	4.7	5.4	6.2					
9			Login forgot password screen	Development	1.0	4.0	5.0	0.7	3.7	4.3	5.0	5.7					
10			Login change password screen	Development	1.0	4.0	5.0	0.7	3.7	4.3	5.0	5.7					
11			Login - Backend Implementation Authentication (JWT)	Development	3.0	5.0	7.0	0.7	5.0	5.7	6.3	7.0					
12			Login - Backend Implementation authorization	Development	3.0	5.0	7.0	0.7	5.0	5.7	6.3	7.0					
13			Login - Backend forgot password small verification	Development	2	4	6	0.7	4.0	4.7	5.3	6.0					
14	2	Main Dashboard Navigation Menu	Navigation Menu frontend	Development	2	2.5	4	0.3	2.7	3.0	3.3	3.7					
15			Navigation Menu Backend (based on image, username)	Development	1.0	2.5	4.0	0.3	2.5	3.0	3.3	4.0					
16	3	Main Dashboard Dropdowns/Filters	Dropdowns & filters frontend	Development	0.5	3.0	5.0	0.5	3.2	3.7	4.3	4.7					
17			Load data to Dropdowns from backend	Development	0.5	2.0	4.0	0.6	2.2	2.7	3.3	3.8					
18	4	Main Dashboard Number Cards	Number Cards frontend (initial load & filter)	Development	1.0	1.0	2.0	0.3	1.2	1.3	1.5	1.7					
19			Number Cards backend (initial load & filter)	Development	2.0	2.5	4.0	0.3	2.7	3.0	3.3	3.7					
20	5	Recruitment Pipeline Stages Horizontal bar chart	chart frontend (initial load & filter)	Development	2.0	2.5	4.0	0.3	2.7	3.0	3.3	3.7					
21			chart backend (initial load & filter)	Development	3.0	5.0	5.5	0.4	4.8	5.2	5.6	6.0					
22	6	Offer Acceptance/ Decline Ratio Doughnut chart	chart frontend (initial load & filter)	Development	4.0	4.5	6.0	0.3	4.7	5.0	5.3	5.7					
23			chart backend (initial load & filter)	Development	4.0	4.5	6.0	0.3	4.7	5.0	5.3	5.7					
24	7	Applications received chart	chart frontend (initial load & weekly filter)	Development	2.0	3.0	4.5	0.4	3.1	3.5	3.9	4.3					
25			chart backend (initial load & weekly filter)	Development	2.0	3.0	4.5	0.4	3.1	3.5	3.9	4.3					
26	9	Application source category wise pie chart	Graph frontend (initial load & filter)	Development	2.0	4.0	6.0	0.7	4.0	4.7	5.3	6.0					
27			Graph backend (initial load & filter)	Development	2.0	4.0	6.0	0.7	4.0	4.7	5.3	6.0					
28	10	Application sources wise pie chart	Graph frontend (initial load & filter)	Development	2.5	4.0	6.0	0.6	4.1	4.7	5.3	5.8					
29			Graph backend (initial load & filter)	Development	2.5	4.0	6.0	0.6	4.1	4.7	5.3	5.8					
30	11	Application Sources filters & search for table	filter frontend	Development	4.0	5.0	7.0	0.5	5.2	5.7	6.2	6.7					
31			filter backend	Development	4.0	5.0	7.0	0.5	5.2	5.7	6.2	6.7					
32	12	Drilldown from Application sources wise pie chart	get records frontend (show data according to the main dashboard)	Development	1.0	2.5	4.0	0.5	2.5	3.0	3.5	4.0					
33	11	Application Sources Table Initial Load	Table frontend	Development	1.5	3.0	4.5	0.5	3.0	3.5	4.0	4.5					
34			Table backend	Development	1.5	3.0	4.5	0.5	3.0	3.5	4.0	4.5					
35	12	Pipeline stages Dropdowns and filters	Filters frontend	Development	5.0	8.0	10.0	0.8	7.8	8.7	9.5	10.3					
36			Filters backend	Development	6.0	8.0	10.0	0.8	7.8	8.7	9.5	10.3					

Figure 62 - Mitra 3P estimation Sheet

- **The important things we should consider while filling estimation sheet**
 1. **Estimate Constraints**

In this part, we have to mention all the problems that we identified at the moment. For example, we have decided to use the ngx-admin angular dashboard template. The template, by default, has a world map. But that was not sufficient to implement our country-wise skill category variation wireframe. Therefore, we decided to use the “leaflet” open source library to draw the maps. But it will take some additional time if that library doesn’t function well after the development. Likewise we have to state our current identified issues in this place. As another example, peopleHR doesn’t have some methods to expose the data through its API. In those situations, we have to think about alternative methods to get the data from the system.

2. **Estimate Risk**

Here we should identify all the risks that will affect the project in the future. It is vital because if a risk occurred while doing the whole project and timeline would be changed, and individual efforts would also be changed according to the situation. For example,

think we are developing our system using a sandbox environment. If the sandbox environment is down for a week, we could not be able to work on the project. Therefore that will directly affect the timeline, and the effort will differ. To avoid this, we should have to mark this in the estimation sheet under the estimation risk row. In some estimation sheets, there is a risk buffer column to put in the extra effort we need to recover in this case. However, in our estimation sheet, it wasn't available.

3. Estimate Assumptions

This part should mention under what assumptions we have designed the system. This is also much important to mention because if the developers cannot meet the deadline, then they can justify it using the assumptions they state in the estimation sheet. For example, the peopleHR system will be available throughout the project is an assumption in the ATS project.

2.4.4.Task 4 – Involved in creating the Project Timeline and Gantt chart

This was a collaborative task assigned to all the developers in the applicant tracking system project. At the beginning of the task, we have to search for a free tool suitable for drawing a Gantt chart with the timeline. There were some software which was named as free but actually not. Because most of them ask to enter a credit card after finishing the drawing. However, finally, I have proposed to the team to use Jira project management tools; hence it is free, and all the developers are using it in the today software industry. The main advantage was that Jira could assign each developer and send a reminder to the respective developer at the same time. Therefore the developer knows that he is assigned to a new task .on the other hand, we can give access to the developer to update the working time on each task while they are working on it. There are some more features that JIRA give for a free account.



Figure 63-ATS Grant chart(due to the space limitation only include first 23 tasks)

2.5. Non-Technical Experience Gained at Mitra Innovation

Communication Skill development

While working on the triton project, I have to record our previous day's work to both team and the client In the morning. At the end of the day, one meeting was there to discuss the difficulties occurring at work. If any help needs, we can raise help at that meeting. So I have the chance to develop my listening and speaking skills.

Presentation Skill development

You must follow the Mitra presentation guideline to create a presentation in Mitra innovation. Only the Mitra documents should be used while presenting anything to the client.

Chapter 3 -Conclusion

3.1. Internship Experience at Mitra Innovation

During the six-month Training period, I was able to work as a back-end developer for a client project and work on an internal project as a designer and developer. From both projects, I got lots of experience in back-end development and a few front-end technologies. In the beginning, I was onboarded to the project called Triton as a Golang back-end developer. Golang was new to me. Therefore Project's Principal Architect Mr. Thilina Herath and My supervisor Mr. deleema Fernando decided to give us three weak study time on the new technology stacks that we are going to use in the project. I had to cover Kubernetes, docker, and Golang within that period. Within my first project, I was able to learn to create Golang REST APIS using microservices architecture, create Swagger API documentations, Golang data seeders, and CSV file reading mechanism, and I especially had the chance to work in A.W.S. services. Learned Industry best practices while working on the project. It was very hard because since we were new to software engineering, most of the concepts we used in the first project were not learned from any of the modules in the faculty. Therefore we have to work day and night to learn and develop the software.

3.2. Self-Assessment on the Internship

An internship at Mitra Innovation is well suitable for computer engineering undergraduates. The reason is you have the chance to work with actual client projects. So while working on those projects, you can identify your weakness. Because while working alone, you eventually get the skills to how to solve issues in code. But If you get help from some other and do it again, once you are stuck, you need help from others. In Mitra innovation whole internship process is different because, in Mitra Innovation, there is no orientation period. So If you recruit as intern today, you will get the chance to work on a client project at the same day you are recruited. I have gotten exposure to the Whole SDLC life cycle. The next thing is Most of the

project Mitra handle is foreign. So you have the chance to participate in all the scrum meetings. Then automatically, your verbal abilities will be up. Most of the developers will help you even if you fail the situation to survive. Because most of the developers are friendly. Therefore only you need to work smart, not work hard.

3.3. Suggestions

My suggestions to future interns who join Mitra Innovation would be to do your best and take the maximum out of every opportunity you get. Create connections throughout the organization, in every division, and do not be limited to your own team. Explore other areas as well. Get the maximum out of the Mitra Innovation Moratuwa Office experience and always try to give your best to the work you do and improve yourself. The company will also benefit by improving yourself, and you will also automatically be benefited. My suggestion to the Department of Computer Engineering, Faculty of Engineering, U.S.J. is to move the industrial training of the Computer Engineering Department to the end of the 4th year as it attracts the leaders in Industry towards the students that we have in our arsenal. And I would also like to suggest to N.A.I.T.A. to digitalize the documentation process, especially in the Computer Engineering/I.T. industry training, as most of the companies are now moving towards a hybrid approach of working and getting digital signatures would be more convenient

List of References

- [1]. Forbes, E., Creating a restful API with Golang. *TutorialEdge*. Available at: <https://tutorialedge.net/golang/creating-restful-api-with-golang/> [Accessed September 19, 2022].
- [2]. Manvar, H., 2021. Securing the application with Kong & Keycloak -101. *Medium*. Available at: <https://faun.pub/securing-the-application-with-kong-keycloak-101-e25e0ae9ec56> [Accessed September 19, 2022].
- [3]. Anon, Build fast, reliable, and efficient software at scale. *Go*. Available at: <https://go.dev/> [Accessed September 19, 2022].
- [4]. Anon, Gorm. *gorm package - gorm.io/gorm - Go Packages*. Available at: <https://pkg.go.dev/gorm.io/gorm#pkg-index> [Accessed September 19, 2022].
- [5]. Anon, 2022. Cloud API management & microservices connectivity platform: KONG. *Kong Inc*. Available at: <https://konghq.com/> [Accessed September 19, 2022].
- [6]. *What is swagger*. [Online]. Available: <https://swagger.io/docs/specification/2-0/what-is-swagger/#:~:text=Swagger%20allows%20you%20to%20describe,beautiful%20and%20interactive%20API%20documentation>. [Accessed: 19-Sep-2022].
- [7]. Swaggo, “Swaggo/SWAG: Automatically generate restful API documentation with swagger 2.0 for go.,” *GitHub*. [Online]. Available: <https://github.com/swaggo/swag>. [Accessed: 19-Sep-2022].

Certification

Name of the Trainee : U.D. Pasindu Dhanushka Uduwela
ID No : 962301231V
Address : 100/128 Padukka Rd. Horana
Institution and Address : Faculty of Engineering (Ratmalana Premises), University of Sri Jayewardenepura, 41 Lumbini Ave, Dehiwala-Mount Lavinia
Programme : Industrial Training(software engineering) period : 6 months
Training Organization and Address : Mitra Innovation (Pvt.) Ltd , 472, Galle Road, Rawatawatta, Moratuwa 10400
Training Period : 2022/02/02 to 2022/08/05

This is to certify that this report is prepared by myself under the training conducted at the above organization



.....
signature of the trainee (with data)

This is to certify that this is prepared by the above trainee under my regular supervision

.....
signature of the Employee with stamp (with data)