



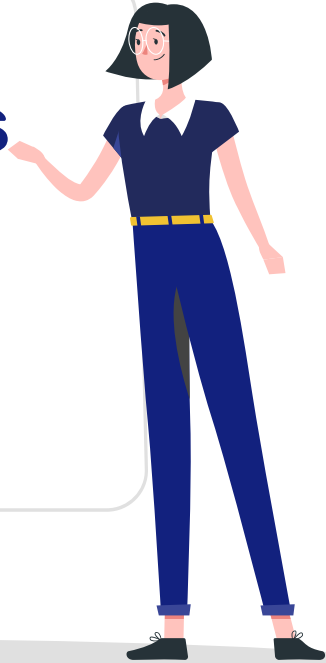
# Ascenda

Loyalty Program API

**Austin Woon, Lee Sherman, Wong Javier,  
Tan Leonard, Tan Brennan, Khoo Ernest**



# **Business Needs & Solution Overview**





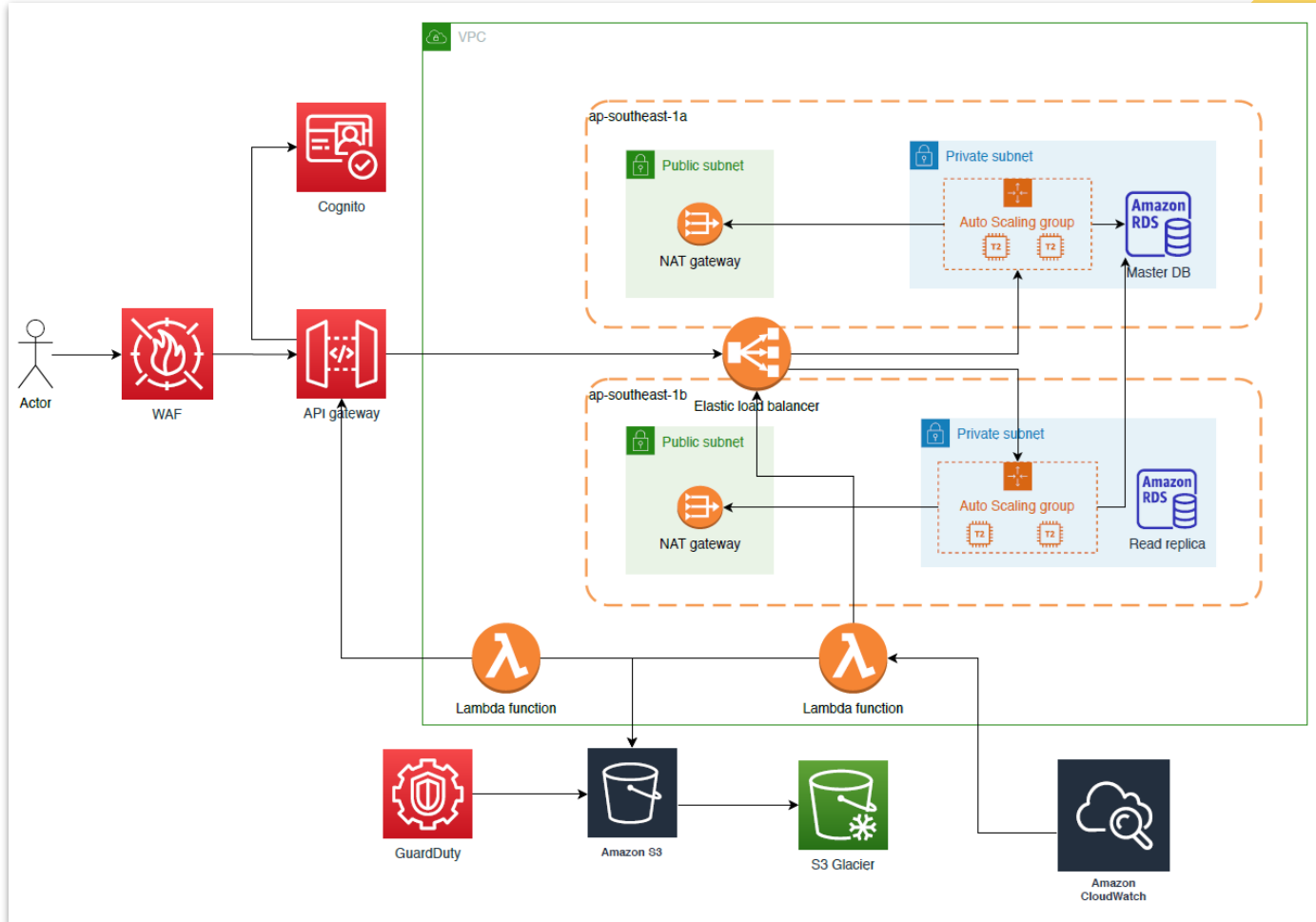
# **Business Needs**

**Loyalty partners are customers of Ascendas and require a solution that will facilitate easy and seamless loyalty currency transfers between loyalty programs**

**Ascendas must provide its customers with an API that will**

- **Display information of loyalty programs on the bank's frontend**
- **Perform loyalty membership validation**
- **Accept & process accrual information for loyalty programs on behalf of banks**
- **Return transaction details when complete and allow querying of transfer details**
- **Perform fulfilment with actual loyalty program, integrating with their specific transfer formats**

# Solution Overview





# **Product Features**

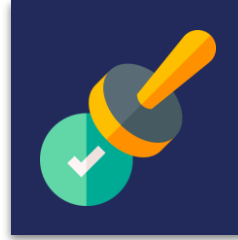
An illustration of a woman with short black hair, wearing a dark blue short-sleeved shirt with a white collar, dark blue trousers with a yellow belt, and black shoes. She is standing on a grey oval shadow and pointing her right hand towards a large, light grey rounded rectangle. Inside the rectangle, the text 'Product Features' is written in a bold, dark blue font. The background is white, with yellow geometric shapes (diagonal lines and squares) in the top-left and bottom-right corners.

# Features Requested



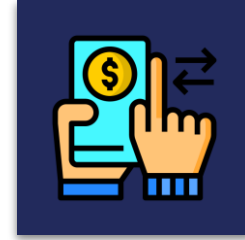
## Loyalty Program API

*Single source of truth* for latest information regarding loyalty programs



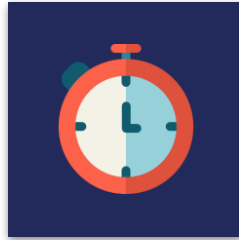
## Membership Validation

*Validate Memberships* of Users belonging to loyalty programs



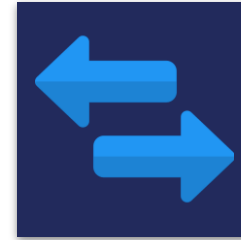
## Accrual Requests

Process *new accrual requests* from Bank



## Transaction Enquiry

Allow banks to *poll* for transaction updates periodically



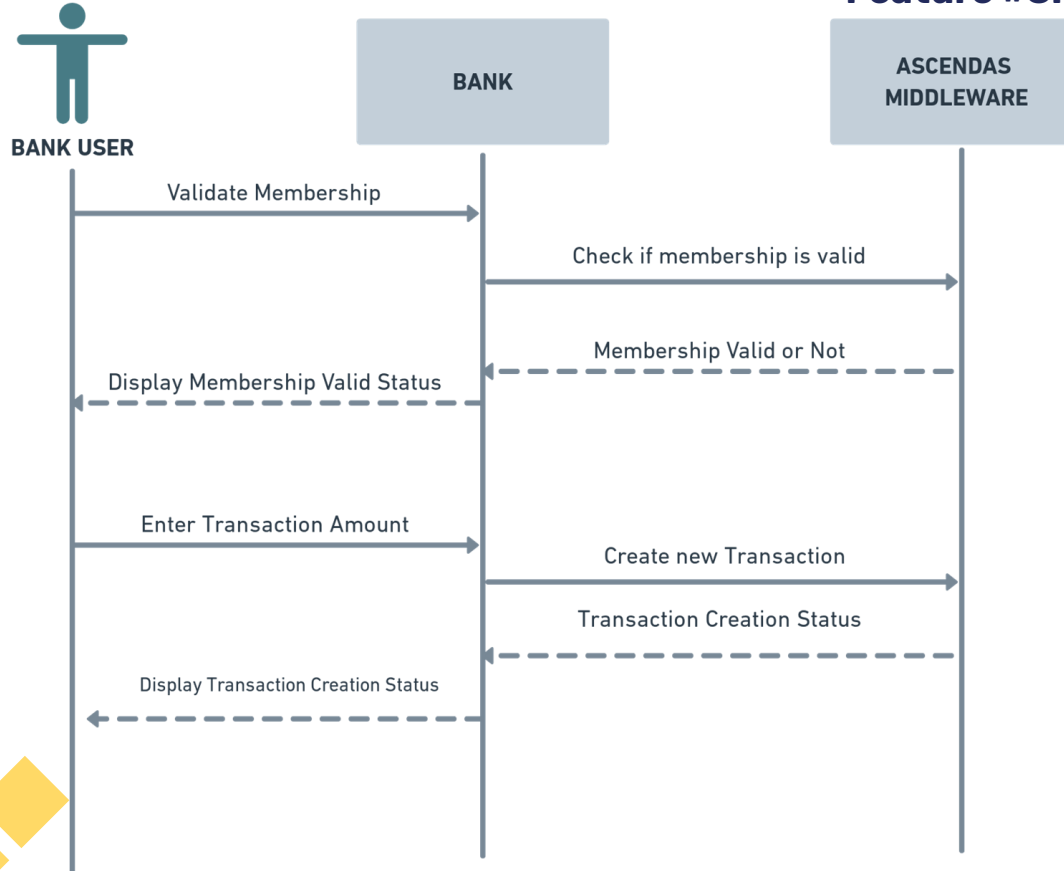
## Transfer Fulfilments

Handle *transfer fulfilments* between Loyalty Partners and Banks

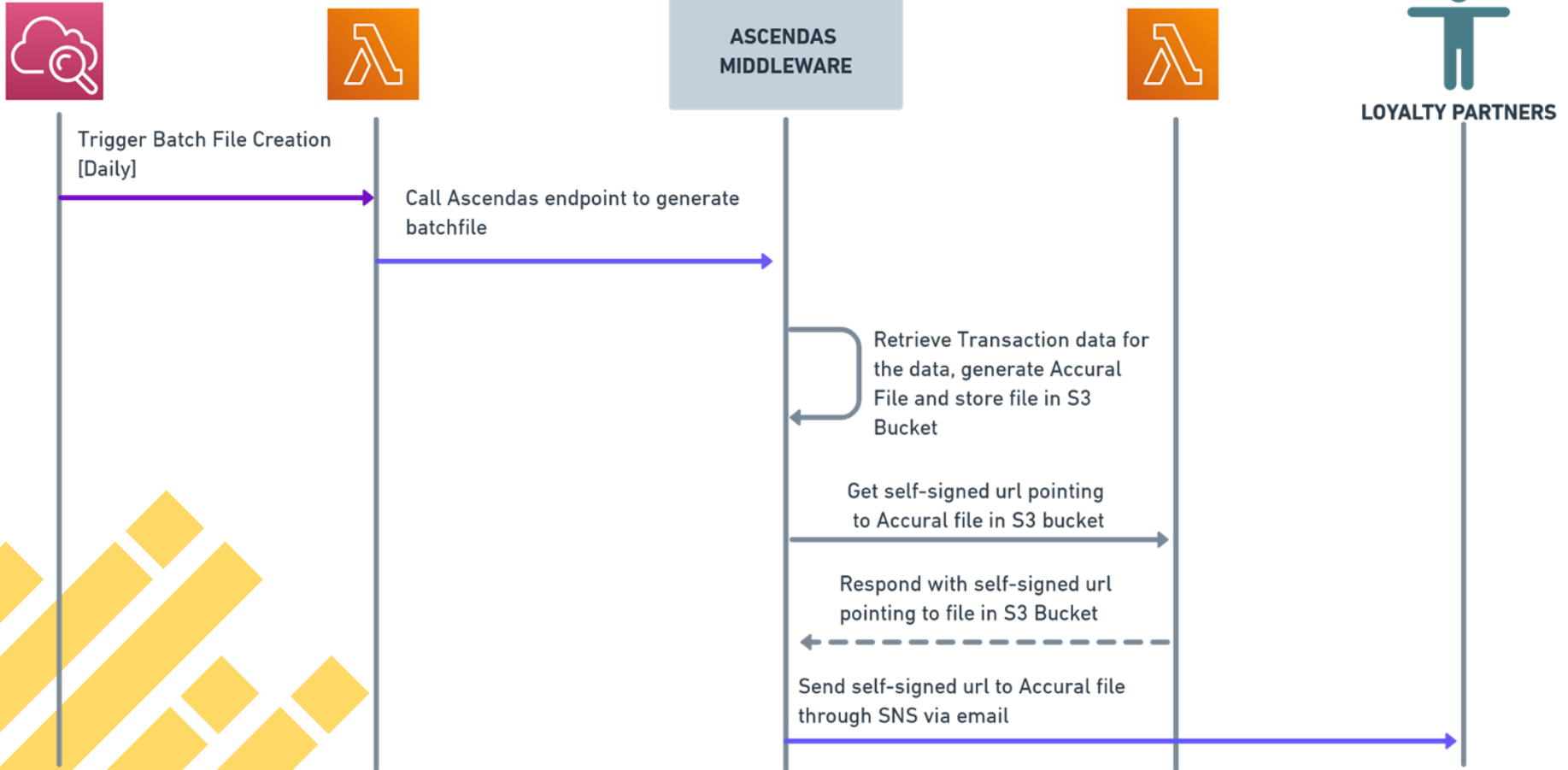
# Feature #1: Loyalty Program API

## Feature #2: Membership Validation

## Feature #3: New Accrual Request

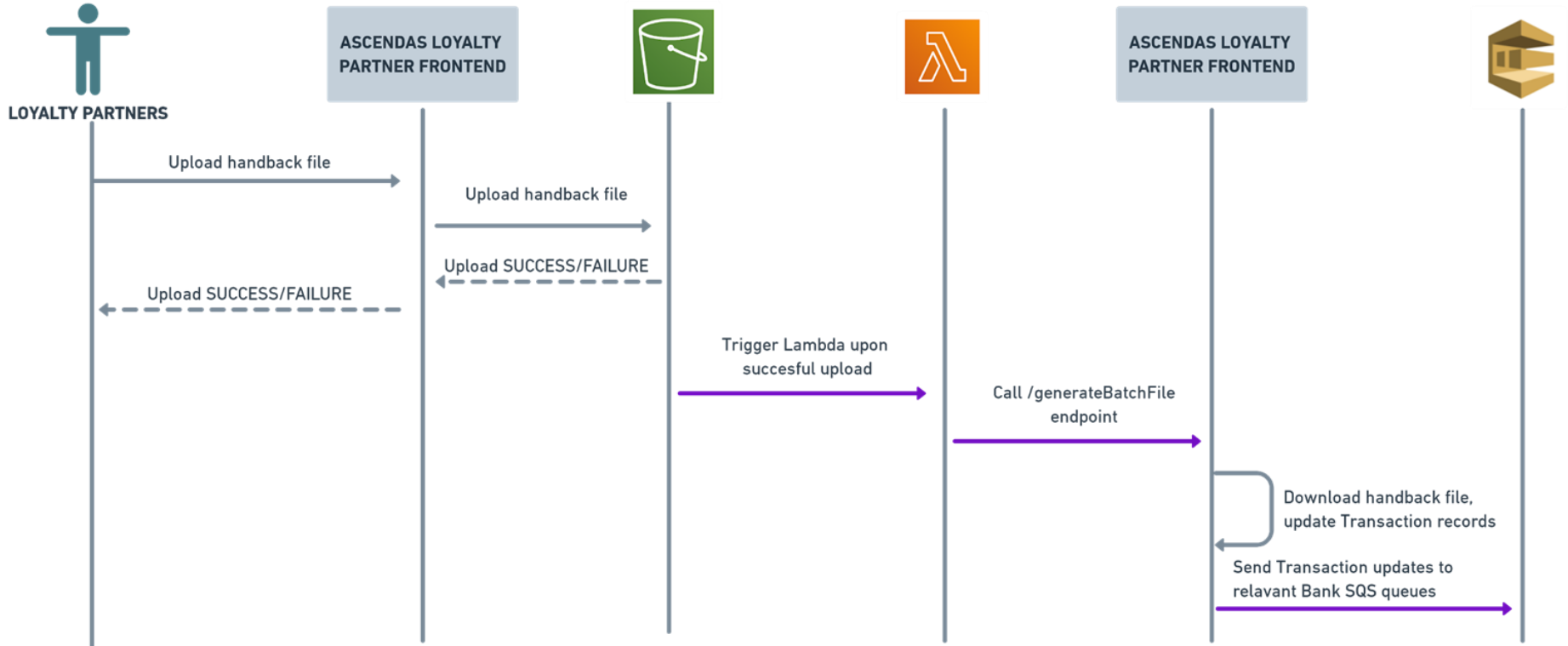


# Feature #5: Transaction Fulfilments

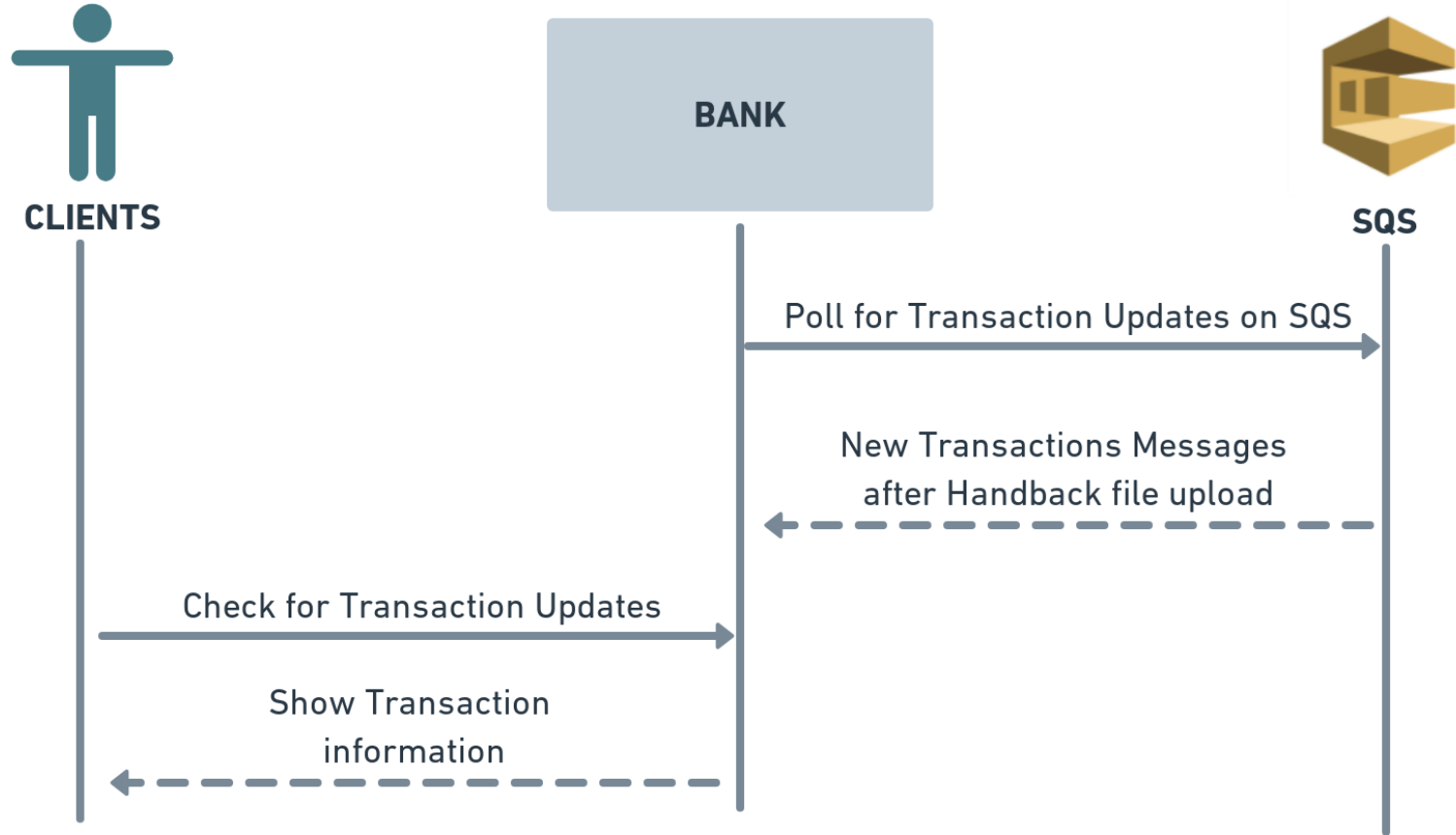


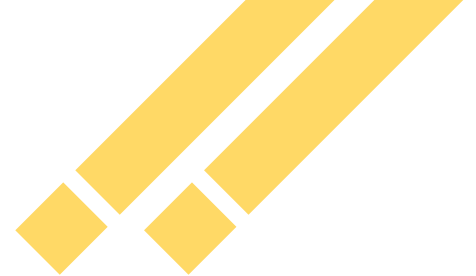


# Feature #5: Transaction Fulfilments



# Feature #4: Transaction Enquiry





**PRODUCT DEMO**

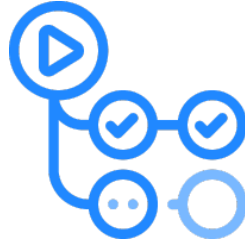


# Maintainability





# Development Process - Pipeline



## Github Actions



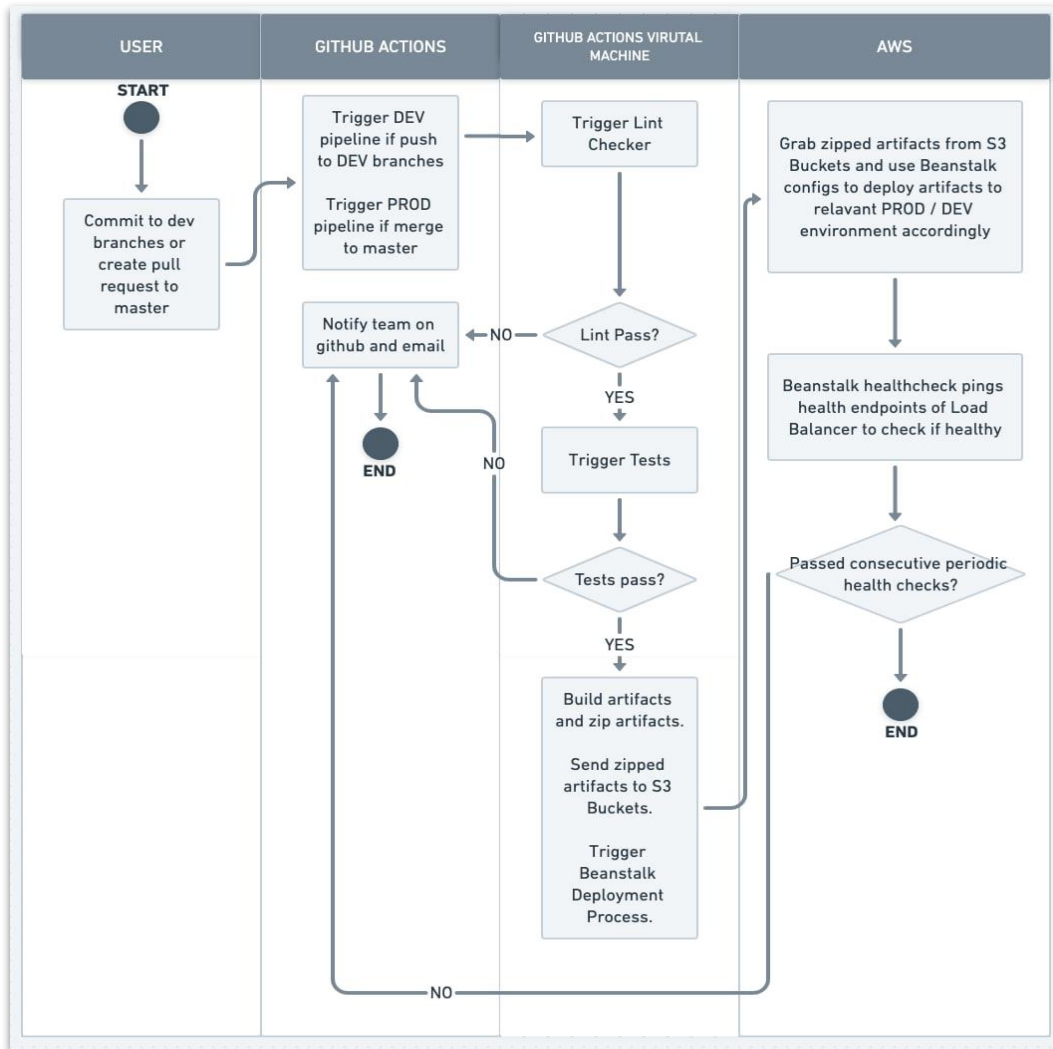
Lint Check for  
Syntax Errors

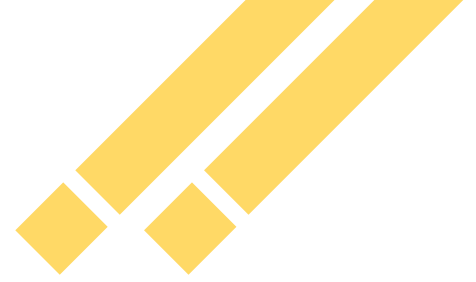


Run all test cases



Deploy on AWS





# **PIPELINE CODE WALKTHROUGH & DEMO**



# Maintainability Design and Implementation



HashiCorp

# Terraform







**Availability**

# Availability Design and



# Implementation - Instance and DB

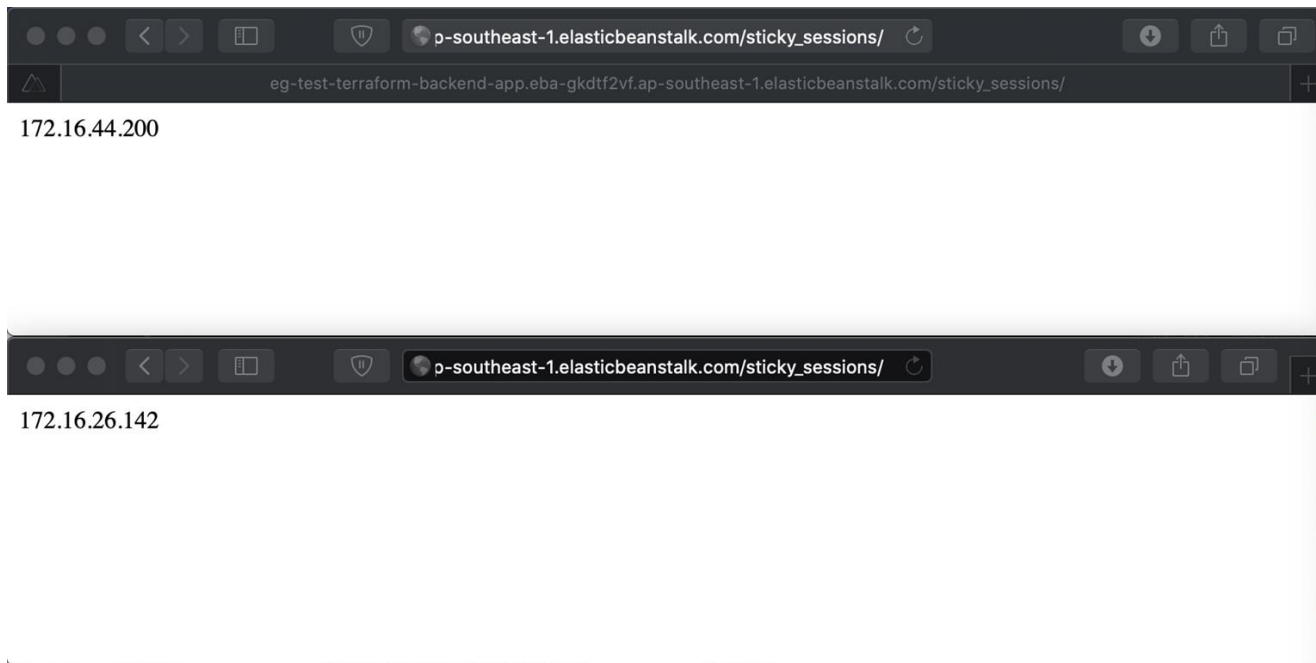
- EC2 instance availability
- RDS availability
- Demo time

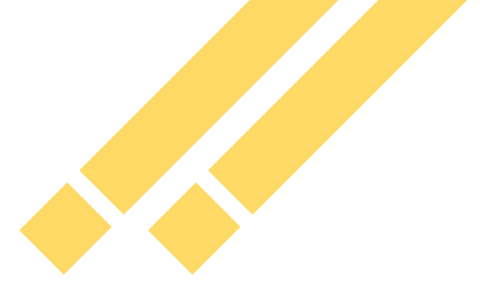




# Availability Design and Implementation - Sticky Sessions

- We did it
- Not very applicable in our scenario





**Security**



# Security Design and Implementation

Feature 1: Ensure application data is sent over an encrypted network tunnel

Connectivity & security		
Endpoint & port	Networking	Security
Endpoint itsa.cjlb3arilazv.ap-southeast-1.rds.amazonaws.com	Availability zone ap-southeast-1b	VPC security groups <a href="#">rds-sg (sg-0593cff6d505ed42e)</a> ( active )
Port 3306	VPC <a href="#">itsa-main2-vpc (vpc-01c86d7583e799403)</a>	Public accessibility No
	Subnet group default-vpc-01c86d7583e799403	<div>Certificate authority rds-ca-2019</div>
	Subnets <a href="#">subnet-011b66e17c3b9b32e</a> <a href="#">subnet-034262533c13e31c7</a> <a href="#">subnet-0e82e659052960d44</a> <a href="#">subnet-025664adf0602ca48</a> <a href="#">subnet-029a884180959f1f4</a>	Certificate authority date Aug 23rd, 2024



# Security Design and Implementation

## Feature 2: Ensuring proper access control between components

Security Groups (1/2) Info

Filter security groups

search: sg-0bcc610785acd29e2 X Clear filters

	Name	Security group ID	Security group name	VPC ID
<input checked="" type="checkbox"/>	itsa-backend-dev8	sg-0717fa10e232af881	awseb-e-bpmm32ygy...	vpc-01c86d7583e799403
<input type="checkbox"/>	itsa-backend-dev8	sg-0bcc610785acd29e2	awseb-e-bpmm32ygy...	vpc-01c86d7583e799403

Type	Protocol	Port range	Source	- optional
HTTP	TCP	80	sg-0bcc610785acd29e2 (awseb-e-bpmm32ygyd-stack-AWSEBLoadBalancerSecurityGroup-71ZJ7HSLUY2T)	-
All traffic	All	All	0.0.0.0/0	-
All traffic	All	All	:::/0	-

ELB Security Groups

Instances (1/7) Info

Filter instances

Instance state: running X Clear filters

	Name	Instance ID	Instance state	Instance type	Status check	Alarm Status	Availability zone
<input checked="" type="checkbox"/>	eg-test-terra...	i-0d240a7788647dc8f	Running	t2.micro	2/2 checks ...	No alarms +	ap-southeast-1b
<input type="checkbox"/>	TO_SSH_IN	i-073744e1164303077	Running	t2.micro	2/2 checks ...	No alarms +	ap-southeast-1a
<input type="checkbox"/>	itsa-backend...	i-0fee4940f06a1597	Running	t2.micro	2/2 checks ...	No alarms +	ap-southeast-1a
<input type="checkbox"/>	eg-test-terra...	i-012195ffa4df9cb4	Running	t2.micro	2/2 checks ...	No alarms +	ap-southeast-1a

Filter rules

Port range	Protocol	Source	Security groups	eg-test-ter...	awseb-e-p...	db-sg
All	All	sg-0db1f6caba50eb538	eg-test-terraform-backend-app	⊙		
80	TCP	sg-0018043e3141929...	awseb-e-paendaepsy-stack-AWSEBS...		⊙	
3306	TCP	sg-051f0a09b02ed3a5e	db-sg-20201107212911093800000...			

EC2 Security Groups



# Security Design and Implementation

## Feature 2: Ensuring proper access control between components

Security Groups (1/3) Info

Filter security groups

search: eg-test-terraform-backend-app X Clear filters

sg-00dc7231u0221d17 - aws-elb-cx2n62zwij-stack-AWSEBLoadBalancerSecurityGroup-F534WZV0DEQN

Details Inbound rules Outbound rules Tags

Inbound rules Edit inbound rules

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	sg-0c883e848cd66609b (aws-elb-cx2n62zwij-stack-AWSEBLoadBalancerSecurityGroup-K4DD7R02GW2U)	-

Security Group Referencing

Security Groups (1/1) Info

Filter security groups

search: sg-0593cff6d505ed42e X Clear filters

<input checked="" type="checkbox"/>	Name	Security group ID	Security group name	VPC ID
<input checked="" type="checkbox"/>	itsa-main2-rds	sg-0593cff6d505ed42e	rds-sg	vpc-01c86d7583e799403

Inbound rules Edit inbound rules

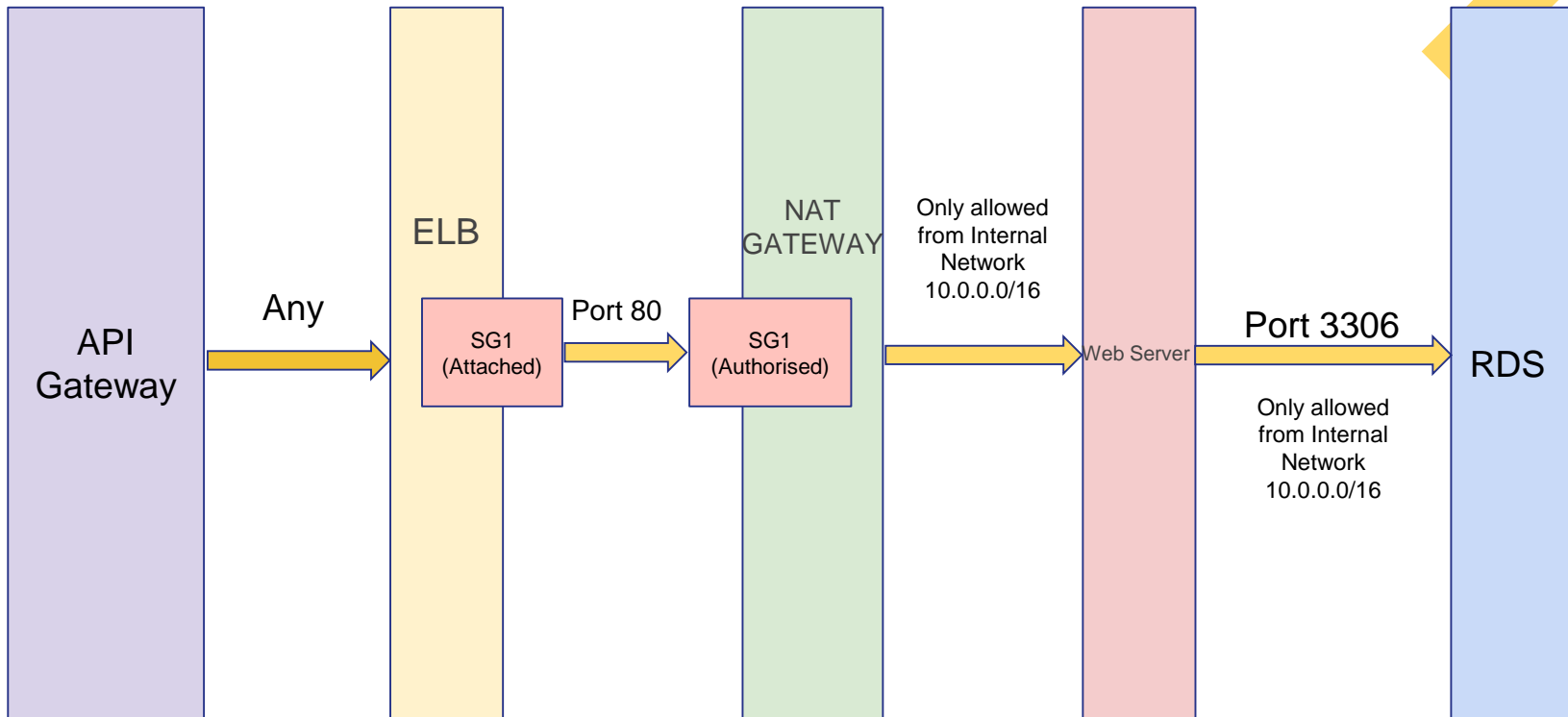
Type	Protocol	Port range	Source	Description - optional
MySQL/Aurora	TCP	3306	10.0.0.0/16	-

RDS Security Group



# Security Design and Implementation

## Feature 2: Ensuring proper access control between components

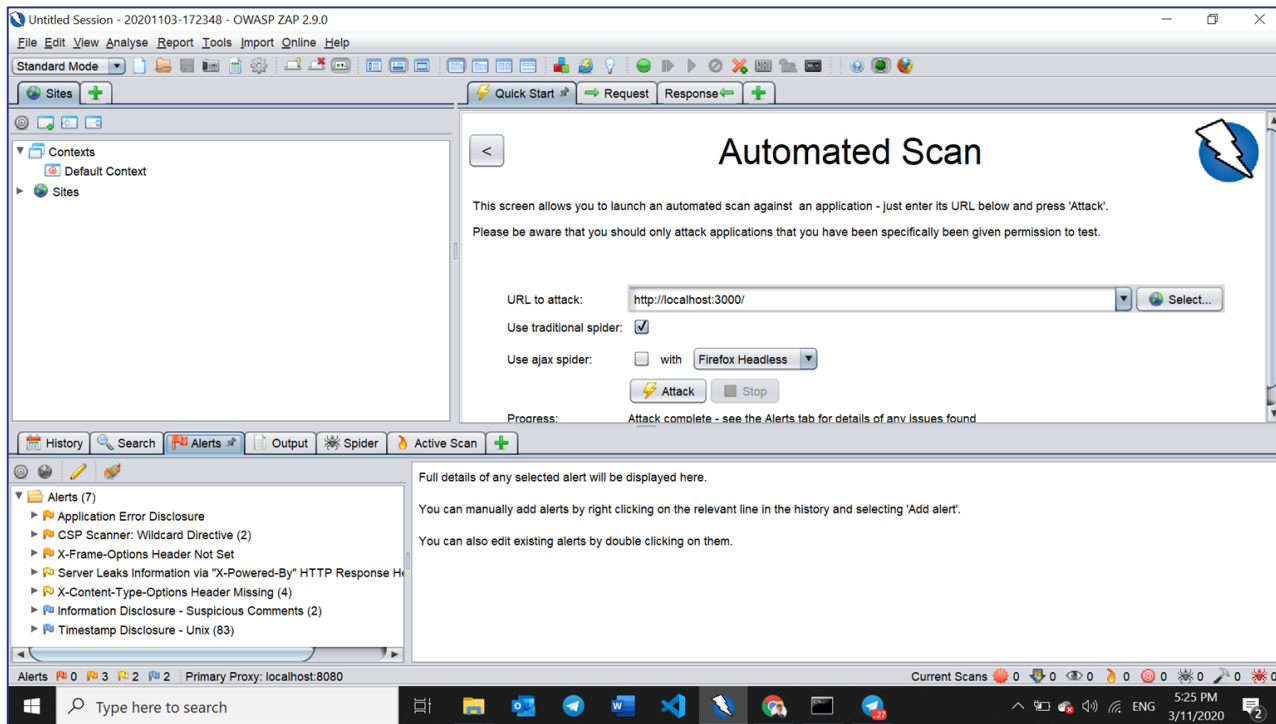






# Security Design and Implementation

## Feature 3: Penetration Testing using OWASP Zap (Localhost)





# Security Design and Implementation

## Feature 3: Penetration Testing using OWASP Zap (Beanstalk)

The screenshot displays the OWASP ZAP web interface. The main window is titled "Automated Scan" and contains instructions for launching a scan. The "URL to attack:" field is populated with "https://loyalty-partner-frontend-static.s3-ap-southeast-1.amazonaws.com/index.html". The "Use traditional spider:" checkbox is checked, and the "Use ajax spider:" checkbox is unchecked. The "Attack" button is highlighted in yellow. The "Progress:" section indicates "Attack complete - see the Alerts tab for details of any issues found".

The bottom panel shows the "Alerts" tab with a list of detected issues. The first alert is "X-Frame-Options Header Not Set", which is highlighted. The details for this alert are shown on the right:

- X-Frame-Options Header Not Set**
- URL: https://loyalty-partner-frontend-static.s3-ap-southeast-1.amazonaws.com/index.html
- Risk: Medium
- Confidence: Medium
- Parameter: X-Frame-Options
- Attack:
- Evidence:
- CWE ID: 16
- WASC ID: 15
- Source: Passive (10020 - X-Frame-Options Header Scanner)
- Description:



# Security Design and Implementation

## Feature 4: 4 Key Vulnerabilities

Asset	Potential Threat/ Vulnerability Pair	Potential Mitigation Controls
Batch/Accrual files stored in Amazon S3	Malicious form of information exposure / Unencrypted data files	Enable server side encryption in us using S3 managed keys (AES-256) or Customer master keys from KMS.



Amazon S3



AWS KMS



# Security Design and Implementation

## Feature 4: 4 Key Vulnerabilities

Amazon S3 > itsa-bank-front-end > Edit default encryption

### Edit default encryption

#### Default encryption

Automatically encrypt new objects stored in this bucket. [Learn more](#)

Server-side encryption

☐ Disable

☒ Enable

Encryption key type

To upload an object with a customer-provided encryption key (SSE-C), use the AWS CLI, AWS SDK, or Amazon S3 REST API.

☒ Amazon S3 key (SSE-S3)

An encryption key that Amazon S3 creates, manages, and uses for you. [Learn more](#)

☐ AWS Key Management Service key (SSE-KMS)

An encryption key protected by AWS Key Management Service (AWS KMS). [Learn more](#)

Cancel

Save changes



# Security Design and Implementation

## Feature 4: 4 Key Vulnerabilities

Asset	Potential Threat/ Vulnerability Pair	Potential Mitigation Controls
Loyalty Partners' Credentials	Server side request forgery (SSRF) / Token based authentication	Add multi-factor authentication (MFA) to the user pool in cognito to protect the identity of your users



**AWS Cognito**



# Security Design and Implementation

## Feature 4: 4 Key Vulnerabilities

aws

Services ▾

ernest.khoo.2018@sis.smu.edu.sg @ 7317-0622-6892 ▾ Singapore ▾ Support ▾

User Pools | Federated Identities

loyaltyPartner

General settings

Users and groups

Attributes

Policies

MFA and verifications

Advanced security

Message customizations

Tags

Devices

App clients

Triggers

Analytics

App integration

App client settings

Domain name

UI customization

Resource servers

### Do you want to enable Multi-Factor Authentication (MFA)?

Multi-Factor Authentication (MFA) increases security for your end users. If you choose 'optional', individual users can have MFA enabled. You can only choose 'required' when initially creating a user pool, and if you do, all users must use MFA. Phone numbers must be verified if MFA is enabled. You can configure adaptive authentication on the Advanced security tab to require MFA based on risk scoring of user sign in attempts. [Learn more about multi-factor authentication.](#)

*Note: separate charges apply for sending text messages.*

☒ Off ☐ Optional ☐ Required

### How will a user be able to recover their account?

When a user forgets their password, they can have a code sent to their verified email or verified phone to recover their account. You can choose the preferred way to send codes below. We recommend not allowing phone to be used for both password resets and multi-factor authentication (MFA). [Learn more.](#)

☒ Email if available, otherwise phone, but don't allow a user to reset their password via phone if they are also using it for MFA

☐ Phone if available, otherwise email, but don't allow a user to reset their password via phone if they are also using it for MFA

☐ Email only

☐ Phone only, but don't allow a user to reset their password via phone if they are also using it for MFA



# Security Design and Implementation

## Feature 4: 4 Key Vulnerabilities

Asset	Potential Threat/ Vulnerability Pair	Potential Mitigation Controls
Ascenda's Database	One possible attack would be a malicious form of information exposure that could exploit the data in the database if it is too openly exposed.	We enforced a permission of least privilege where each specific bank is authorised to only poll from a specific SQS queue using IAM credentials. <b>[Implemented through SQS &amp; IAM]</b>



AWS IAM



**amazon**  
SQS



# Security Design and Implementation

## Feature 4: 4 Key Vulnerabilities

Amazon SQS > Queues > dbs-poll.fifo

**dbs-poll.fifo** Edit Delete Purge Send and receive messages

**Details** [Info](#)

Name	Type	ARN
dbs-poll.fifo	FIFO	arn:aws:sqs:ap-southeast-1:731706226892:dbs-poll.fifo
Encryption	URL	Dead-letter queue
Disabled	https://sqs.ap-southeast-1.amazonaws.com/731706226892/dbs-poll.fifo	Disabled

**Access policy** Edit

Define who can access your queue. [Info](#)

```
{
  "Version": "2008-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__owner_statement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::731706226892:root"
      },
      "Action": "SQS:*",
      "Resource": "arn:aws:sqs:ap-southeast-1:731706226892:dbs-poll.fifo"
    }
  ]
}
```





# Security Design and Implementation

## Feature 4: 4 Key Vulnerabilities

Asset	Potential Threat/ Vulnerability Pair	Potential Mitigation Controls
Users' Credentials, Membership & Transaction information	One possible attack would be a malicious form of information exposure that could exploit the unencrypted RDS data and snapshots.	Ensure RDS data and snapshots are encrypted. This ensures data at rest and snapshots are encrypted.





# Security Design and Implementation

## Feature 4: 4 Key Vulnerabilities

<b>Snapshots</b>	
Automated backups	Option group
Reserved instances	default:mysql-8-0
Proxies	Zone
	ap-southeast-1b
Subnet groups	KMS key ID
Parameter groups	arn:aws:kms:ap-southeast-1:731706226892:key/1a1bcdaf-0ed5-4ca7-a089-8a5b80e5c0e6
Option groups	Source region
Events	N/A
Event subscriptions	

Instance			
Configuration	Instance class	Storage	Performance Insights
DB instance id	Instance class	<b>Encryption</b>	Performance Insights
itsa	db.t2.small	Enabled	enabled
Engine version	vCPU	KMS key	No
8.0.20	1	<a href="#">aws/rds</a>	



# Security Design and Implementation

## Feature 5: Other Security Designs: AWS WAF



AWS WAF

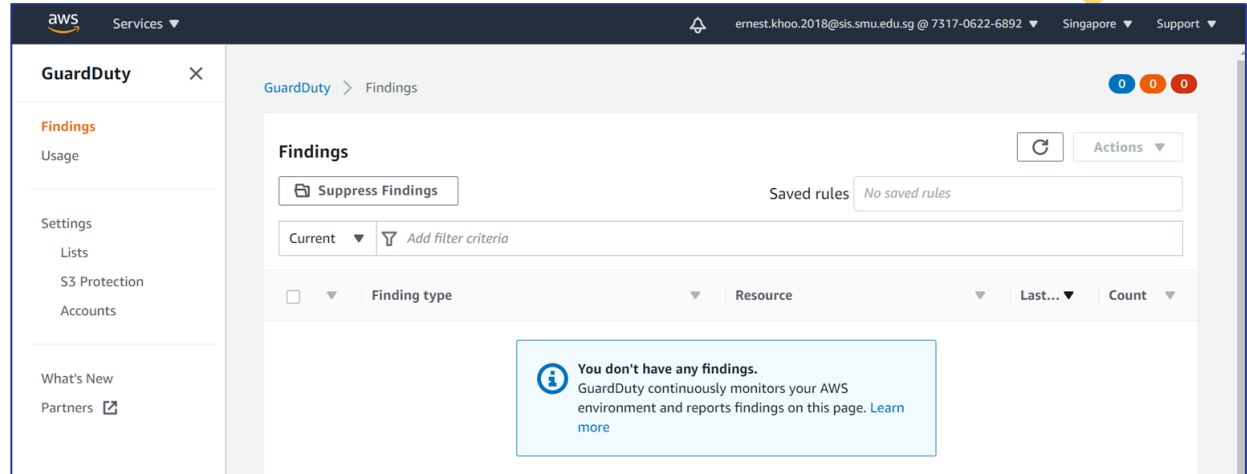
<b>Amazon IP reputation list</b> This group contains rules that are based on Amazon threat intelligence. This is useful if you would like to block sources associated with bots or other threats.	25	<input checked="" type="checkbox"/> Add to web ACL <input type="checkbox"/> Set rules action to count
<b>Anonymous IP list</b> This group contains rules that allow you to block requests from services that allow obfuscation of viewer identity. This can include request originating from VPN, proxies, Tor nodes, and hosting providers. This is useful if you want to filter out viewers that may be trying to hide their identity from your application.	50	<input type="checkbox"/> Add to web ACL
<b>Core rule set</b> Contains rules that are generally applicable to web applications. This provides protection against exploitation of a wide range of vulnerabilities, including those described in OWASP publications.	700	<input checked="" type="checkbox"/> Add to web ACL <input type="checkbox"/> Set rules action to count

<b>Linux operating system</b> Contains rules that block request patterns associated with exploitation of vulnerabilities specific to Linux, including LFI attacks. This can help prevent attacks that expose file contents or execute code for which the attacker should not have had access.	200	<input checked="" type="checkbox"/> Add to web ACL <input type="checkbox"/> Set rules action to count
<b>PHP application</b> Contains rules that block request patterns associated with exploiting vulnerabilities specific to the use of the PHP, including injection of unsafe PHP functions. This can help prevent exploits that allow an attacker to remotely execute code or commands.	100	<input type="checkbox"/> Add to web ACL
<b>POSIX operating system</b> Contains rules that block request patterns associated with exploiting vulnerabilities specific to POSIX/POSIX-like OS, including LFI attacks. This can help prevent attacks that expose file contents or execute code for which access should not be allowed.	100	<input type="checkbox"/> Add to web ACL
<b>SQL database</b> Contains rules that allow you to block request patterns associated with exploitation of SQL databases, like SQL injection attacks. This can help prevent remote injection of unauthorized queries.	200	<input checked="" type="checkbox"/> Add to web ACL <input type="checkbox"/> Set rules action to count



# Security Design and Implementation

## Feature 5: Other Security Designs: Amazon GuardDuty





# Security Design and Implementation

## Feature 5: Other Security Designs: API Gateway



**Amazon API  
Gateway**

Launchpad GET Week 6 Labs No Environment

Week 6 Labs Examples 0 BUILD

GET https://2njpo4jo9a.execute-api.ap-southeast-1.amazonaws.com/PROD/users Send Save

Params Authorization Headers (17) Body Pre-request Script Tests Settings Cookies

Headers 9 hidden

KEY	VALUE	DESCRIPTION	*** Bulk Edit Presets
<input checked="" type="checkbox"/> x-api-key	[REDACTED]		
<input type="checkbox"/>			
<input type="checkbox"/>			

Body Cookies Headers (11) Test Results Status: 200 OK Time: 194 ms Size: 834 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "userId": 1,
4     "firstName": "michael",
5     "lastName": "jackson",
6     "partnerCode": "dbs",
7     "pointBalance": "9999994369"
8   },
9   {
10    "userId": 2,
11    "firstName": "freddie",
12    "lastName": "mercury",
13    "partnerCode": "dbs",
14    "pointBalance": "10000000000"
15  },
16  {
```

# Performance





# Performance Design and Implementation

Global Performance: Ensure requests time < 200ms

```
validateTimer: 83.2529296875 ms
▶ Fetch finished loading: POST "https://ji7f79xnwd.execute-api.ap-southeast-1.amazonaws.com/test/validatemembership".
Success: ▶ {rewardsAmount: undefined}
accrualTimer: 182.02685546875 ms
▶ Fetch finished loading: POST "https://ji7f79xnwd.execute-api.ap-southeast-1.amazonaws.com/test/newaccrual".
```

# Cache Implementation



**Amazon API  
Gateway**

With API Gateway caching, we can reduce the number of calls and also improve the latency of requests to our API



# Cache Implementation



## Without Cache

```
▶ 1000 Fetch finished loading: GET "<URL>".  
successfully called 100 calls  
successfully called 200 calls  
successfully called 300 calls  
successfully called 400 calls  
successfully called 500 calls  
successfully called 600 calls  
successfully called 700 calls  
successfully called 800 calls  
successfully called 900 calls  
0.8368699999817618
```

## With Cache

```
▶ 1000 Fetch finished loading: GET "<URL>".  
successfully called 100 calls  
successfully called 200 calls  
successfully called 300 calls  
successfully called 400 calls  
successfully called 500 calls  
successfully called 600 calls  
successfully called 700 calls  
successfully called 800 calls  
successfully called 900 calls  
0.6512600000169186
```

Average time taken was calculated from 1000 requests

# Parallel Execution

```
Nov 7 10:21:49 ip-10-0-1-251 web: [2020-11-07 10:21:49 +0000] [4911] [INFO] Handling signal: term
Nov 7 10:21:49 ip-10-0-1-251 web: [2020-11-07 10:21:49 +0000] [4911] [INFO] Worker exiting (pid: 4911)
Nov 7 10:21:50 ip-10-0-1-251 web: [2020-11-07 10:21:50 +0000] [4853] [INFO] Shutting down: Master
Nov 7 10:21:50 ip-10-0-1-251 web: [2020-11-07 10:21:50 +0000] [5367] [INFO] Starting gunicorn 20.0.4
Nov 7 10:21:50 ip-10-0-1-251 web: [2020-11-07 10:21:50 +0000] [5367] [INFO] Listening at: http://127.0.0.1:8000 (5367)
Nov 7 10:21:50 ip-10-0-1-251 web: [2020-11-07 10:21:50 +0000] [5367] [INFO] Using worker: threads
Nov 7 10:21:50 ip-10-0-1-251 web: [2020-11-07 10:21:50 +0000] [5423] [INFO] Booting worker with pid: 5423
Nov 7 11:06:01 ip-10-0-1-251 web: [2020-11-07 11:06:01 +0000] [5367] [INFO] Handling signal: term
Nov 7 11:06:02 ip-10-0-1-251 web: [2020-11-07 11:06:02 +0000] [5423] [INFO] Worker exiting (pid: 5423)
Nov 7 11:06:02 ip-10-0-1-251 web: [2020-11-07 11:06:02 +0000] [5367] [INFO] Shutting down: Master
Nov 7 11:06:03 ip-10-0-1-251 web: [2020-11-07 11:06:03 +0000] [6337] [INFO] Starting gunicorn 20.0.4
Nov 7 11:06:03 ip-10-0-1-251 web: [2020-11-07 11:06:03 +0000] [6337] [INFO] Listening at: http://127.0.0.1:8000 (6337)
Nov 7 11:06:03 ip-10-0-1-251 web: [2020-11-07 11:06:03 +0000] [6337] [INFO] Using worker: threads
Nov 7 11:06:03 ip-10-0-1-251 web: [2020-11-07 11:06:03 +0000] [6393] [INFO] Booting worker with pid: 6393
Nov 7 11:14:55 ip-10-0-1-251 web: [2020-11-07 11:14:55 +0000] [6337] [INFO] Handling signal: term
Nov 7 11:14:55 ip-10-0-1-251 web: [2020-11-07 11:14:55 +0000] [6393] [INFO] Worker exiting (pid: 6393)
Nov 7 11:14:55 ip-10-0-1-251 web: [2020-11-07 11:14:55 +0000] [6337] [INFO] Shutting down: Master
Nov 7 11:14:55 ip-10-0-1-251 web: [2020-11-07 11:14:55 +0000] [3208] [INFO] Starting gunicorn 20.0.4
Nov 7 11:14:55 ip-10-0-1-251 web: [2020-11-07 11:14:55 +0000] [3208] [INFO] Listening at: http://127.0.0.1:8000 (3208)
Nov 7 11:14:55 ip-10-0-1-251 web: [2020-11-07 11:14:55 +0000] [3208] [INFO] Using worker: threads
Nov 7 11:14:55 ip-10-0-1-251 web: [2020-11-07 11:14:55 +0000] [3267] [INFO] Booting worker with pid: 3267
Nov 7 11:28:30 ip-10-0-1-251 web: [2020-11-07 11:28:30 +0000] [3208] [INFO] Handling signal: term
Nov 7 11:28:30 ip-10-0-1-251 web: [2020-11-07 11:28:30 +0000] [3267] [INFO] Worker exiting (pid: 3267)
Nov 7 11:28:31 ip-10-0-1-251 web: [2020-11-07 11:28:31 +0000] [3208] [INFO] Shutting down: Master
Nov 7 11:28:31 ip-10-0-1-251 web: [2020-11-07 11:28:31 +0000] [3782] [INFO] Starting gunicorn 20.0.4
Nov 7 11:28:31 ip-10-0-1-251 web: [2020-11-07 11:28:31 +0000] [3782] [INFO] Listening at: http://127.0.0.1:8000 (3782)
Nov 7 11:28:31 ip-10-0-1-251 web: [2020-11-07 11:28:31 +0000] [3782] [INFO] Using worker: threads
Nov 7 11:28:31 ip-10-0-1-251 web: [2020-11-07 11:28:31 +0000] [3835] [INFO] Booting worker with pid: 3835
Nov 7 12:35:42 ip-10-0-1-251 web: [2020-11-07 12:35:42 +0000] [3782] [INFO] Handling signal: term
Nov 7 12:35:43 ip-10-0-1-251 web: [2020-11-07 12:35:43 +0000] [3835] [INFO] Worker exiting (pid: 3835)
Nov 7 12:35:43 ip-10-0-1-251 web: [2020-11-07 12:35:43 +0000] [3782] [INFO] Shutting down: Master
Nov 7 12:35:44 ip-10-0-1-251 web: [2020-11-07 12:35:44 +0000] [4934] [INFO] Starting gunicorn 20.0.4
Nov 7 12:35:44 ip-10-0-1-251 web: [2020-11-07 12:35:44 +0000] [4934] [INFO] Listening at: http://127.0.0.1:8000 (4934)
Nov 7 12:35:44 ip-10-0-1-251 web: [2020-11-07 12:35:44 +0000] [4934] [INFO] Using worker: threads
Nov 7 12:35:44 ip-10-0-1-251 web: [2020-11-07 12:35:44 +0000] [4989] [INFO] Booting worker with pid: 4989
Nov 7 13:05:47 ip-10-0-1-251 web: [2020-11-07 13:05:47 +0000] [4934] [INFO] Handling signal: term
Nov 7 13:05:48 ip-10-0-1-251 web: [2020-11-07 13:05:48 +0000] [4989] [INFO] Worker exiting (pid: 4989)
Nov 7 13:05:48 ip-10-0-1-251 web: [2020-11-07 13:05:48 +0000] [4934] [INFO] Shutting down: Master
Nov 7 13:05:48 ip-10-0-1-251 web: [2020-11-07 13:05:48 +0000] [5708] [INFO] Starting gunicorn 20.0.4
Nov 7 13:05:48 ip-10-0-1-251 web: [2020-11-07 13:05:48 +0000] [5708] [INFO] Listening at: http://127.0.0.1:8000 (5708)
Nov 7 13:05:48 ip-10-0-1-251 web: [2020-11-07 13:05:48 +0000] [5708] [INFO] Using worker: threads
Nov 7 13:05:48 ip-10-0-1-251 web: [2020-11-07 13:05:48 +0000] [5763] [INFO] Booting worker with pid: 5763
Nov 7 14:30:15 ip-10-0-1-251 web: [2020-11-07 14:30:15 +0000] [5708] [INFO] Handling signal: term
Nov 7 14:30:15 ip-10-0-1-251 web: [2020-11-07 14:30:15 +0000] [5763] [INFO] Worker exiting (pid: 5763)
Nov 7 14:30:15 ip-10-0-1-251 web: [2020-11-07 14:30:15 +0000] [5708] [INFO] Shutting down: Master
Nov 7 14:30:15 ip-10-0-1-251 web: [2020-11-07 14:30:15 +0000] [7013] [INFO] Starting gunicorn 20.0.4
Nov 7 14:30:15 ip-10-0-1-251 web: [2020-11-07 14:30:15 +0000] [7013] [INFO] Listening at: http://127.0.0.1:8000 (7013)
Nov 7 14:30:15 ip-10-0-1-251 web: [2020-11-07 14:30:15 +0000] [7013] [INFO] Using worker: threads
Nov 7 14:30:15 ip-10-0-1-251 web: [2020-11-07 14:30:15 +0000] [7068] [INFO] Booting worker with pid: 7068
```

1

Implemented gunicorn workers

2

Handle multiple requests in parallel

3

Improve our performance drastically



**Thanks!**

**Do You Have Any  
Questions?**