SINGAPORE INSTITUTE OF TECHNOLOGY

School of Infocomm Technology

Applied Artificial Intelligence, BSc (Hons)

INF2006 CLOUD COMPUTING AND BIG DATA

Group 1

TEAM MEMBERS	STUDENT ID
LEO EN QI VALERIE	2202795
OSAMA RASHEED KHAN	2203385
SERI HANZALAH BTE HANIFFAH	2201601
TEO XUANTING	2202217
TIAN YUE XIAO, BRYON	2201615

Date: 23 February 2024

YouTube Link: https://youtu.be/WgTNxeCbPv0

Submitted as part of the requirement for INF2006 Cloud Computing and Big Data

ALVIN TOONG SOONG CHAN, P.H.D

TABLE OF CONTENTS

INTRODUCTION

In today's digital era, cloud computing has revolutionised the way businesses manage their IT infrastructure which offers scalability, cost efficiency and flexibility. This report presents the design and implementation of a high-availability web application, the Expense Tracking System (ETS). The ETS will be hosted on Amazon Web Services (AWS). The ETS aids in simplifying financial management by providing an intuitive interface for both personal and collective expense tracking. The ETS contains features such as a real-time dashboard, easy expense entry, comprehensive management and insightful reporting which embodies a user-centric approach to financial oversight. The system is built with a focus on high availability and streamlined user experience which aims to redefine how users interact with their financial data in the cloud environment.

ASSUMPTIONS

1. Assuming that our target market is small to medium-sized enterprises (SMEs), the Expense Tracking System (ETS) will be for companies with less than 250 users. This is to allow efficient resource allocation and ensure that the system is optimised to provide better user experiences.

PHASE 01: DESIGN

1.1 Requirement Analysis

Scalability

- The Expense Tracker System (ETS) should support a user base of less than 250 users with the potential to scale up as the number of users grows.
- The ETS should be able to handle an increase in transaction volume without degradation in performance.
- The ETS should be able to scale out resources during peak usage times.

Fault Tolerance

- The ETS must maintain a continuous operation with no single point of failure.
- The ETS should be designed to handle potential system failures gracefully to ensure that the users still have access to their data.
- The ETS needs to have mechanisms that automatically recover from a fault without manual intervention.

Performance

- The ETS should provide a responsive user experience with quick load times, even under high load.
- The ETS should ensure fast processing of transactions and real-time updates of expense records.
- The ETS needs to maintain a consistent performance as the number of concurrent users increases.

Security

- As financial data are sensitive, the ETS must incorporate robust security measures. This includes data encryption and secure access controls.
- User authentication should be strong, possibly incorporating multi-factor authentication.
- The ETS must ensure that user data is not susceptible to unauthorised access or breaches.

Disaster Recovery

- The ETS should include a disaster recovery plan to handle data loss scenarios.
- Regular backups should be scheduled and the ability to quickly restore operations after a disaster should be tested.
- The architecture of the ETS should be designed to replicate data across different geographical locations to prevent data loss due to regional failures.

1.2 Cloud Service Model Selection

The Cloud Service Model chosen for the Expense Tracker System is Platform-as-a-Service (PaaS). The various advantages and disadvantages are shown in Table 1.

TABLE 1. ADVANTAGES AND DISADVANTAGES OF PAAS

PLATFORM AS A SERVICE (PaaS)	
ADVANTAGES	**DISADVANTAGES**
Collaborative Development **Provides a shared software development environment for users to access to all the tools they need in the ETS.**	**Platform / Vendor lock in** **The proprietary technologies and tools from AWS cloud system prevent users from migrating their services to other vendors (Google Cloud Service, Azure) easily**
Additional Development Capabilities **Users can purchase additional capabilities, frameworks, tools, packages for the ETS.**	**Security and data privacy** **Financial data needs to be uploaded to the remote servers and storage in the hybrid cloud. Therefore, may encounter potential network security issues.**
Automated testing and deployment **PaaS enables spin-up and automate the ETS development process, testing and production.**	**Complexity** **Users are limited to specific functionality, performance and integrations offered by SaaS**

Business Viewpoint

PaaS offers a competitive edge in terms of time-to-market. This means that businesses can quickly adapt to the changing market conditions, launch new features and respond to customer feedback with minimal delay. By providing a platform with an existing framework, Paas reduces deployment time which allows businesses to deploy applications rapidly. The agility in deployment is critical for maintaining relevance and responding to market needs.

Secondly, as cost efficiency plays an important factor, PaaS eliminates the need for extensive capital investment in infrastructure, which is particularly advantageous for SMEs with budget considerations. The operational cost savings are substantial as well because PaaS includes infrastructure maintenance, security and scalability which would otherwise require additional staff and resources.

Furthermore, the PaaS model supports business continuity through its robust disaster recovery and fault tolerance capabilities. This ensures that the ETS remains operational which is vital for financial applications where data integrity and availability directly influence business operations and decision making.

Technical Viewpoint

The requirement for scalability is well met by PaaS which offers elastic scalability. As user demand increases and decreases, the resources can be dynamically allocated or unallocated which is essential for managing the ETS's workload as it grows, supporting less than 250 users to potentially more.

PaaS offers a key advantage in fault tolerance compared to IaaS and SaaS. IaaS gives users control over the infrastructure, but it requires the users to manually set up fault tolerance, which can be complex, resource-intensive and time-consuming. Even though SaaS provides full-stack service, it offers limited customisation for specific fault-tolerance needs. For PaaS however, it abstracts the complexity of the infrastructure, providing built-in fault tolerance that ensures high availability without the overhead of managing it. This makes PaaS the ideal cloud service model for the ETS since it aligns with the need for a resilient system that maintains high availability with less hands-on management, allowing the focus to remain on application functionality and user experience.

In terms of performance, PaaS environments are optimised for load balancing and it can handle the real-time processing of transactions efficiently, which is critical for an application that is dealing with financial data.

1.3 Cloud Deployment Strategy

The hybrid cloud deployment model will be used for the Expense Tracking System (ETS). The cloud deployment strategy was chosen based on the following factors:

- Scalability and Flexibility,
- Security,
- Cost Management,
- Business Continuity
- Cloud Operation and Migration

TABLE 2. FACTORS AFFECTING HYBRID CLOUD DEPLOYMENT CHOICE

FACTORS	DESCRIPTIONS
Scalability and Flexibility	The hybrid cloud deployment model allows the ETS for automatic scalability of cloud resources during demand spikes. Critical for handling any sudden ETS usage surge without compromising computational performances. The flexibility allows for the ETS to be always available when needed.
Security and Compliance	The private subnet group contains the RDS database of the financial data. Sensitive financial data are stored and processed in the private cloud of the ETS. Users can also benefit from broader functionalities of the public cloud service for less sensitive operations.
Cost Management	In a hybrid cloud system, the AWS load balancer and auto-scaler enable dynamic scaling up or down as needed. Therefore, reducing unnecessary financial expenditures for maintaining the on-premises cloud infrastructure when not in use.
Business Continuity	In the event of a failure or outage in the cloud environment, a hybrid cloud deployment model offers a more robust framework for disaster recovery by diversifying computational resources across public and private networks, compared to purely private or public cloud deployment models. This hybrid cloud approach helps prevent complete service disruption of the ETS.
Cloud Operation and Migration	Even though the hybrid cloud deployment model is not as flexible as the multi-cloud deployment model in terms of vendor lock-in, the hybrid cloud model allows for a more straightforward and simplified approach to data migration. Several migration techniques can be

	chosen, such as gradual, selective or bi-directional migration. **Gradual Migration**: Involves a phased approach to migrating financial data in the ETS. Enables the users to move workload gradually from on-premise infrastructures to the hybrid cloud. **Selective Migration:** Based on compliance and performance requirements, users can selectively choose which sensitive data in the ETS to migrate to the cloud. This allows for maintaining control over critical financial data. **Bi-directional Migration**: Hybrid cloud supports bi-directional data migration, allowing for data in the ETS to be moved to the cloud and back to on-premises as needed.
Innovation and Modernisation	AWS offers a wide range of advanced cloud services such as AWS Glue and Amazon SageMaker. Through the integration of such services with the ETS, users can gain deeper insights into their financial data and spending patterns to make data-driven decisions.

Therefore, a hybrid cloud deployment model is chosen based on the factors detailed above. The model combines the data security of a private cloud with the scalability and flexibility of the public cloud, supporting the operational demands and security compliance of the ETS. Amazon Web Services (AWS) also offers significant support for hybrid cloud deployment architectures via Amazon RDS on VMware, which is also used for this ETS.

1.4 Cloud Storage and Database

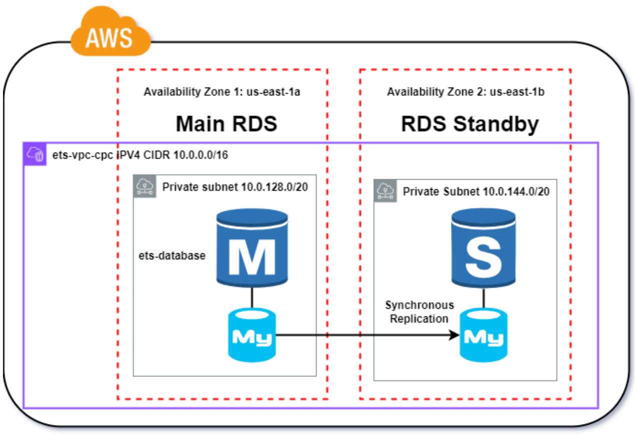

Figure 1. Desired ETS RDS Setup

The ETS is structured as a two-tier cloud web application, featuring both a public and a private subnet layer. Figure 1 displays the setup of the ETS Relational Database System (RDS) within the private subnets. Ideally, the RDS setup spans multiple availability zones for relational data.

MySQL has been chosen for the relational database because the ETS requires the handling and management of structured data, such as transactions, budgets and financial records. The predefined tabular format of relational databases is well suited to reduce the risk of data input anomalies and ensure consistency. MySQL is open-source and one of the most used relational database management systems, thus a cost-effective selection for the ETS. While MySQL can effectively handle significant amounts of structural data input, the database management system also provides the tools to scale up when there is an increase in demands of the ETS.

AWS has been selected as the cloud storage service provider for the reasons outlined in Table 4:

- Market Leadership and Maturity
- Global Infrastructure
- Cost Management Tools
- Customisation and Control

TABLE 4. ADVANTAGES OF AWS CLOUD PROVIDER SERVICES

ADVANTAGES	DESCRIPTIONS
Market Leadership and Maturity	Amongst AWS, Google Cloud Platform (GCP) and Microsoft Azure, AWS has been the leading cloud service provider on the market. With a proven track record and broad customer base, AWS offers more extensive and mature cloud services.
Global Infrastructure	AWS has the largest global infrastructure as compared to GCP and Azure, thus having more availability zones and data centres. Thus offering lower latency, more redundancy and better compliance requirements for the ETS.
Cost Management	AWS offers tools for cost management and optimisation. AWS Cost Explorer is used for this ETS to monitor the financial aspects of the hybrid cloud expenditure more effectively.
Customisation and Control	AWS Identity and Access Management (IAM) is used to control access to the AWS EC2 virtual instances.

Our goal was to design a multi-AZ RDS configuration, but due to the limited permissions of our AWS learner accounts, we were restricted to implementing a basic RDS instance setup without any DB instance standby instead. Figure 1 illustrates what our intended multi-AZ RDS arrangement would have resembled.

1.5 High Availability and Secure Configuration

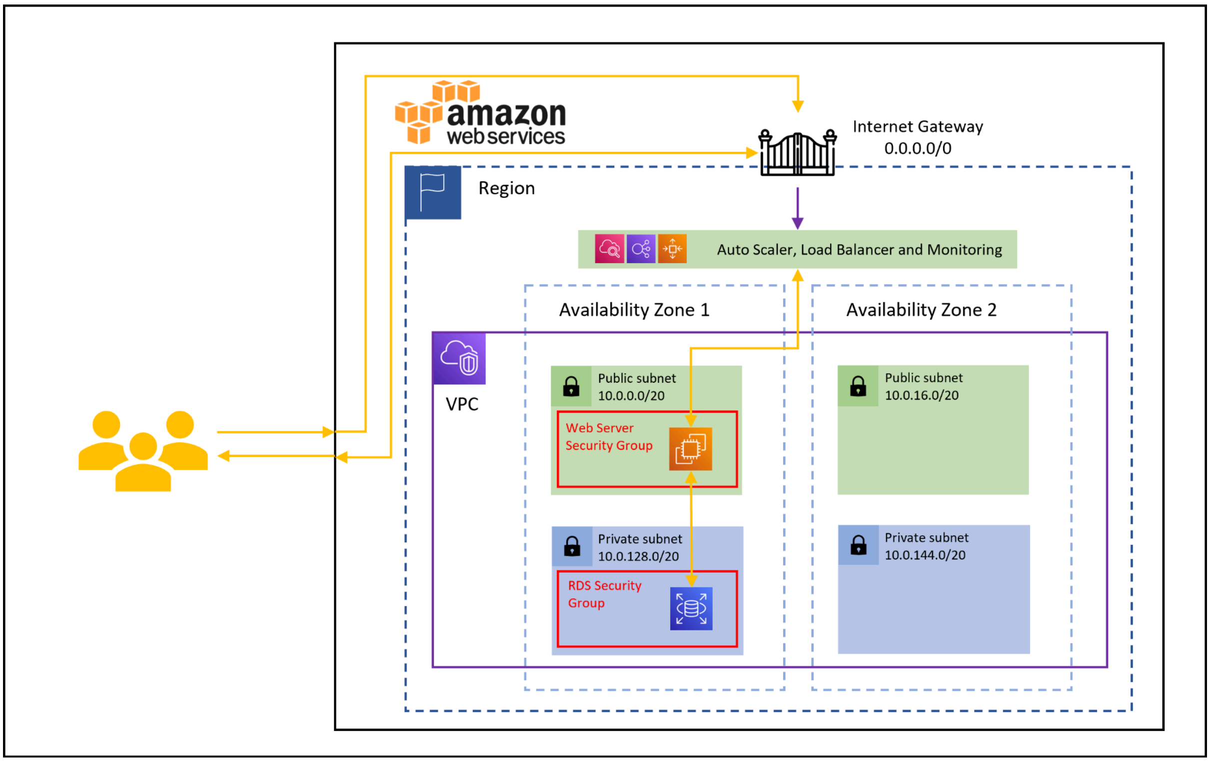

Figure 2. ETS Cloud Architecture Framework

1.5.1 High Availability

- **Multiple AZ**: The hybrid cloud is deployed across two Availability Zones (AZ). This ensures high availability and continued operations using the resources in the other AZ if one AZ is unexpectedly unavailable.
- **Auto Scaling**: The AWS Auto Scaling feature helps to automatically adjust the resource capacity to maintain a steady and predictable cloud performance at the lowest cost. This ensures the ETS can handle increased traffic without the need for manual intervention.

1.5.2 Fault Tolerance

- **Database Replication:** Having multiple AZs allows for database replication. The secondary database acts as a standby and an immediate replacement of the primary database should it fail. Thus, minimising server downtime.
- **Data Backups**: In the event of a potential failure, fault tolerance is maintained due to regular database snapshots and backups. Thus, ensuring that all data can be restored as soon as possible.

1.5.3 Security

- **Security Groups**: Both the RDS instances and the web server are protected inside by their respective security groups. The security groups act as virtual firewalls to ensure only authorised access from the inbound and outbound traffic.
- **Private and Public Subnets**: A default Amazon Virtual Private Clouds (VPC) with a CIDR of `172.31.0.0/16` and distinct public and private subnets isolates the DB

layer from the internet. Only the web servers in the public subnet are exposed to the internet, while the databases are securely hidden in the private subnet.

1.5.4 Load Balancing

- **Amazon Elastic Load Balancing (ELB)**: The ELB automatically distributes incoming traffic across the Amazon EC2 Virtual Machine (VM) instances in multiple AZs.
- **Performance Monitoring:** AWS CloudWatch is used to monitor computational performance and potential bottlenecks in the traffic. An alert system is implemented in the AWS CloudWatch when the CPU usage reaches above the specified threshold.

Based on Figure 2, the architecture ensures that the ETS is capable of handling traffic surges while providing consistent and reliable cloud storage services for the users.

PHASE 02: IMPLEMENTATION

2.1 AWS Setup

2.1.1 Creating Instances

- **AMI and Instance Type Selection:** Selected Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type, and t2.micro instance type to balance stability, security and cost-effectiveness. The t2.micro instance type was chosen because of its low cost and its general purpose instance type.

- **Instance Naming and Security:** The instance is named “ETS RDS Web Server” to facilitate easy identification and management. The public security groups were configured to only allow inbound traffic from SSH, HTTPS and HTTP. This helps to protect against unauthorised data access.

- **Storage Allocation: 8GB** is allocated for storage to ensure sufficient space for the application, logs and data. This helps to optimise operational efficiency and manage costs effectively. In a scenario where storage expansion is needed, the storage can be manually adjusted respectively to ensure the scalability of the ETS.

- **Addition of New Servers using AMI Launch Templates:** Shorten the configuration process and ensure consistency in configuration information across instances.

- **Verification:** The verified instance is created by accessing the web server through the public IP address. This ensures that the instance is properly configured and is externally accessible.

2.1.2 Setting Up Networking

- **VPC Creation:** A VPC named “ets-vpc-cpc” with an IPV4 CIDR block of `10.0.0.0/16` was created to facilitate isolated cloud resources.

- **Subnet Configuration:** 4 subnets were created across each availability zone; **`us-east-1a`** and **`us-east-1b`**. Each of the availability zones contains a private and public subnet. This gives a structured network layer that allows resources to be placed in different network segments for organisational and security purposes.

- **Internet Gateway and NAT Gateways:** The Internet Gateway “ets-vpc-igw” is used to enable instances in the public subnet to communicate with the internet, allowing for inbound and outbound traffic. NAT Gateways “ets-vpc-nat-public1-us-east-1a” and “ets-vpc-nat-public1-us-east-1b” are deployed to allow instances in the private subnets to access the internet for updates and outbound traffic while keeping them secure from inbound traffic.

- **Route Tables:**
  - **Public Route Table**: A route table named "ets-vpc-rtb-public" is associated with the public subnets, containing a route that directs internet-bound traffic to the Internet Gateway.
  - **Private Route Tables**: Two private route tables, "ets-vpc-rtb-private1-us-east-1a" and "ets-vpc-rtb-private1-us-east-1b," are associated with the private subnets in their respective availability zones. Each private route table contains a route that directs internet-bound traffic to the corresponding NAT Gateway.

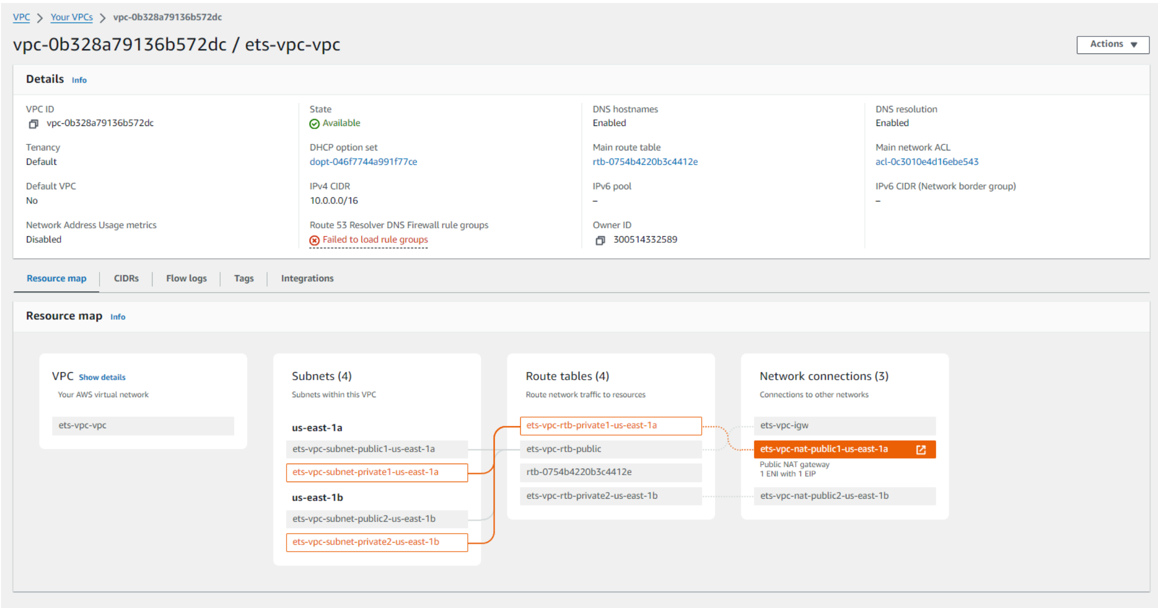

Figure 3. VPC Route tables

2.1.3 Configuring Security Measures

- **Security Groups:** The ETS Web Security Group was set to have inbound traffic for SSH, HTTP, HTTPS, MySQL/Aurora from the Internet Gateway `0.0.0.0/0` to allow access for users to view and add their expenses.

TABLE 5. WEB SECURITY INBOUND RULES

SOURCE	PROTOCOL	TYPE	IP VERSION	PORT RANGE
0.0.0.0/0	TCP	HTTP	IPV4	80
0.0.0.0/0	TCP	HTTPS	IPV4	443
0.0.0.0/0	TCP	MYSQL/Aurora	IPV4	3306
0.0.0.0/0	TCP	SSH	IPV4	22

- **Database Security:** The ETS DB Security Group was set to have inbound traffic for only MySQL/Aurora through the ETS Web Security Group. This assures that only MySQL/Aurora traffic is coming through the ETS Web Security Group as an additional security measure.

TABLE 6. DATABASE SECURITY INBOUND RULES

SECURITY GROUP RULE	PROTOCOL	TYPE	PORT RANGE
sgr-0b57716d4d42eef11	TCP	MYSQL/Aurora	3306

- **Verification and Testing:** The configurations, particularly the security group rules, were validated through connectivity tests between instances, the database, and the internet. This ensures unhindered communication while protecting against unauthorised access. This process includes copying the security group's DNS into HTTP.

2.2 Web Application Deployment

The Expense Tracking System (ETS) was deployed on the AWS EC2 instances. High availability was achieved due to leveraging multiple instances across different availability zones within the region. This ensures that the ETS remains accessible even when one of the zones is facing disruptions.

2.3 Database Implementation

The database was set up with a db.t3.micro class using MySQL Engine with the ETS DB Security Group. It is assigned to the Availability Zone "**`us-east-1a`**" and linked to the ETS RDS Web Server.

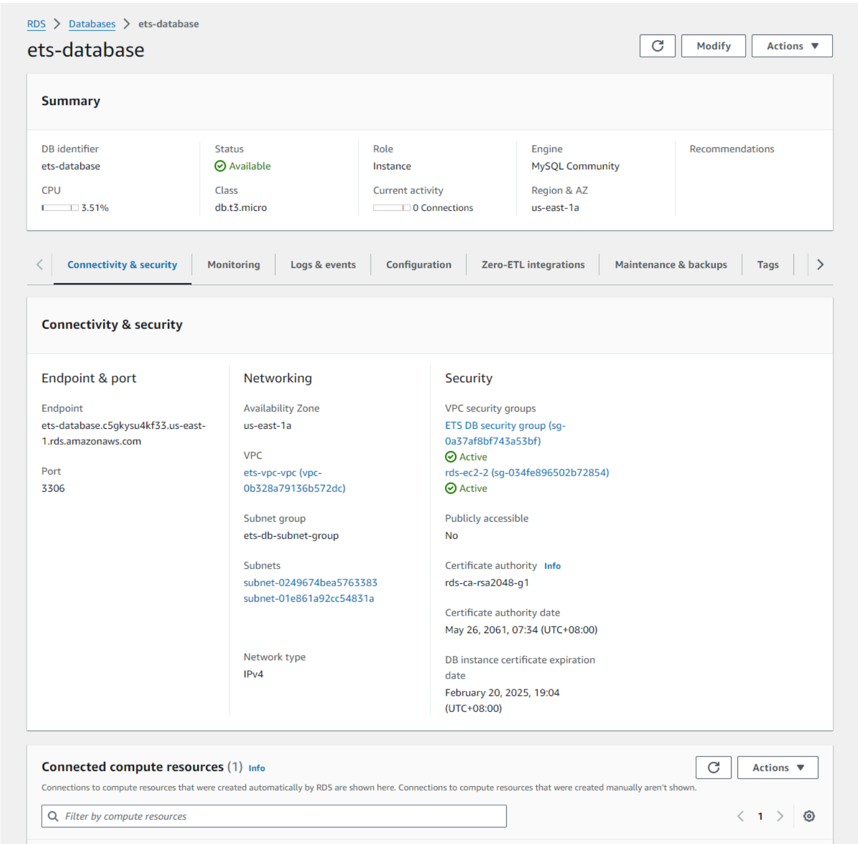

Figure 4. ETS Database Connectivity

2.3.1 Set Up and Configure Database

- **Amazon RDS Utilisation:** MySQL is used as the main relational database (RDS) for managing the structured data input of the ETS.

- **Backup Strategy:** To ensure data recoverability in the event of data loss, Amazon RDS has automated backup services. Ideally, the main RDS database (DB) also has a DB instance standby.

2.3.2 Ensure Data Persistence

Ideally, the ETS employs a multi-AZ database system to ensure data persistence and high availability, as compared to a single-AZ DB system. Table 5 highlights the comparison between single-AZ and multi-AZ database systems, as well as the benefits of employing multi-AZ database standby instances.

TABLE 7. ADVANTAGES OF MULTI-AZ DB STANDBY INSTANCES

ADVANTAGES	SINGLE-AZ	MULTI-AZ
Additional read capacity	Limited to primary	**1 standby:** Standby DB instance is a passive failover target for high availability **2 standbys**: 2 Standby DB instances act as failover targets and serve read traffic
Automatic failover duration		**1 standby**: 60 - 120 secs **2 standbys**: <35 secs
Lower latency for transaction commits		**2 standbys:** Up to 2X faster transaction commits compared to just 1 standby
Higher resiliency to AZ outage	Risk data loss (RPO up to 5 mins) in the event of an AZ failure	**1 standby**: Workload will automatically failover to the up-to-date standby **2 standbys**: 2 Standby DB instances act as failover targets and serve read traffic
Lower jitter for transaction commits	No jitter optimisation	**1 standby:** Sensitive to write path impairments **2 standbys:** Use the 2-of-3 write quorum, insensitive to up to 1 impaired write path

2.3.3 Data Management Capabilities

Secondary database: Preferably, a secondary database is established as a DB instance standby, thereby eliminating any I/O freezes and minimising latency spikes during AWS system backups. In the event of a failover from the primary MySQL relational database, the standby will seamlessly assume the role of the new primary database. Therefore, ensuring high availability uptime and fault tolerances in the Amazon RDS.

2.4 Load Balancing

As stated in section 1.5.4, Amazon ELB was employed in the ETS to distribute incoming traffic across multiple instances for optimal performance.

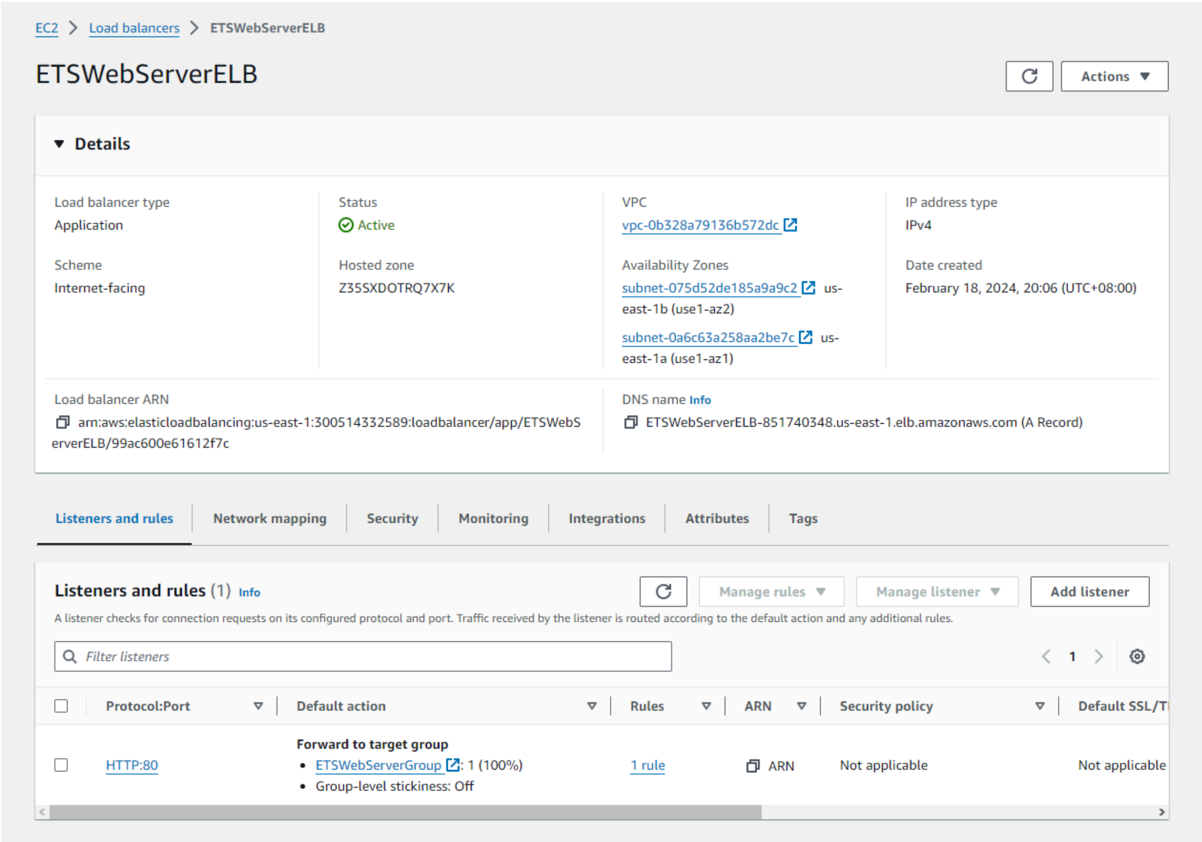

Figure 5. ETS Web Server ELB

2.5 Monitoring and Auto-Scaling

AWS CloudWatch was used to monitor the performance of the ETS. It tracks metrics like CPU utilisation, latency and request counts. Based on the metrics, auto-scaling policies were set up using EC2 Auto Scaling, this ensures that the number of EC2 instances are horizontally scaled up or down automatically to maintain the desired CPU performance levels for each of the instances.

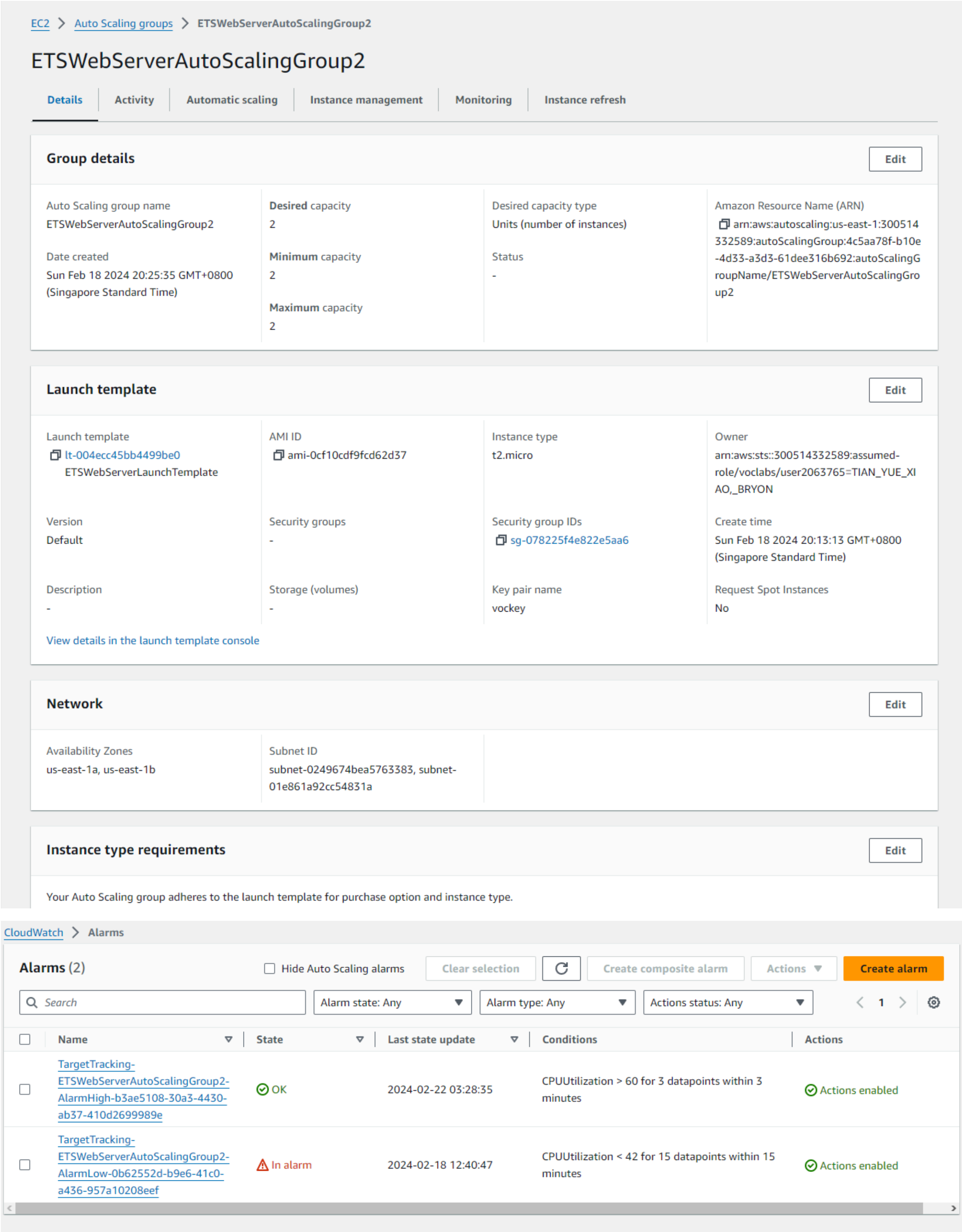

Figure 6. ETS Autoscaler

PHASE 03: ANALYSIS

3.1 Performance Evaluation & Scalability Assessment

Amazon CloudWatch, Amazon EC2 Auto Scaling and JMeter were used to evaluate the performance and the scalability of the Expense Tracking System (ETS).

A load test was simulated using APache JMeter to simulate 8000 users simultaneously with a short time delay registering an account, logging in, adding budget and expense, viewing monthly expenses, navigating to the dashboard and logging out. The JMeter script allows for sending and receiving HTTP/HTTPS requests without the presence of a browser, where the parameters of the load tests can be specified using environment variables. In this context, requests are sent to sign-up.php to register an account, set-budget.php to set a monthly budget, add_expense.php to add an expense and monthly_detailed.php to retrieve the monthly expenses.

Key indicators such as CPU utilisation are monitored via the AWS CloudWatch automatic dashboard to assess performance and analytics along with the EC2 instances generated.

A sharp spike in CPU utilisation was observed as shown in Figure 9. This triggers the Auto Scaling Group to create new instances and shut them down after the user traffic cuts down. It was also noted that the incoming requests reached about 8000 requests at the peak load.

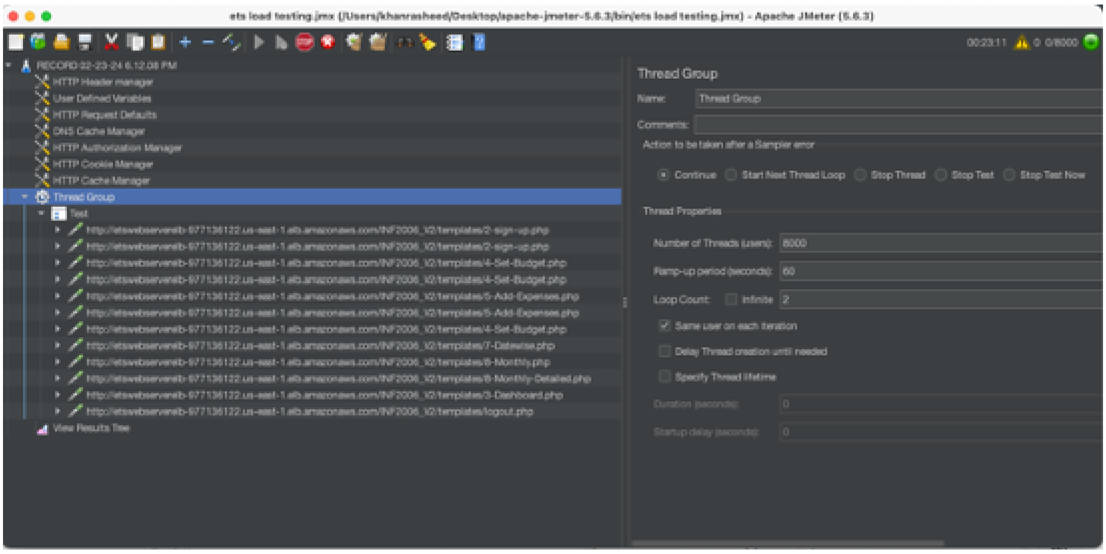

Figure 7. Apache JMeter Load Testing setup

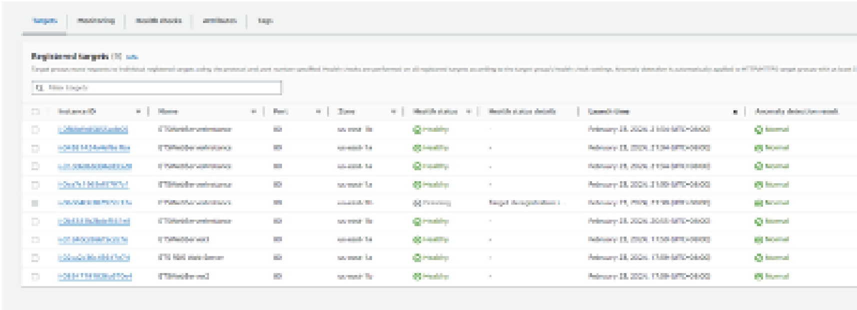

Figure 8. EC2 Instances

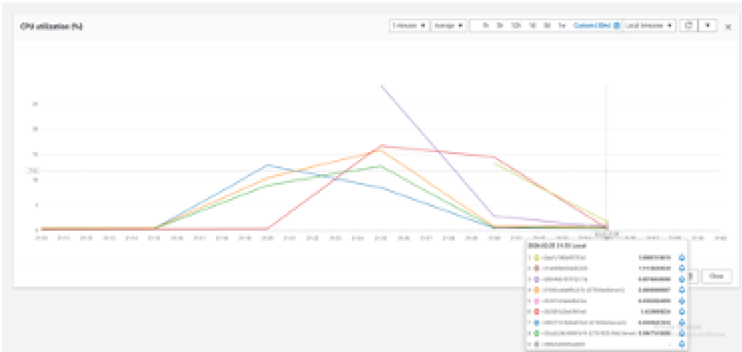

Figure 9. CPU Utilisation Graph

Status	Description	Cause
Successful	Launching a new EC2 instance: i-067c61bc525c6597d	At 2024-02-23T12:55:16Z a monitor alarm TargetTracking-WebServerAutoScalingGroup-AlarmHigh-7e607729-4fc2-491f-87f1-603592f99d32 in state ALARM triggered policy Web Server Scaling Policy changing the desired capacity from 1 to 6. At 2024-02-23T12:55:19Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 6.
Successful	Launching a new EC2 instance: i-099b4d32c351781af	At 2024-02-23T12:55:16Z a monitor alarm TargetTracking-WebServerAutoScalingGroup-AlarmHigh-7e607729-4fc2-491f-87f1-603592f99d32 in state ALARM triggered policy Web Server Scaling Policy changing the desired capacity from 1 to 6. At 2024-02-23T12:55:19Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 6.
Successful	Launching a new EC2 instance: i-0620f878fdad76f57	At 2024-02-23T12:55:16Z a monitor alarm TargetTracking-WebServerAutoScalingGroup-AlarmHigh-7e607729-4fc2-491f-87f1-603592f99d32 in state ALARM triggered policy Web Server Scaling Policy changing the desired capacity from 1 to 6. At 2024-02-23T12:55:19Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 6.
Successful	Launching a new EC2 instance: i-04a8f07b1a2cf2293	At 2024-02-23T12:55:16Z a monitor alarm TargetTracking-WebServerAutoScalingGroup-AlarmHigh-7e607729-4fc2-491f-87f1-603592f99d32 in state ALARM triggered policy Web Server Scaling Policy changing the desired capacity from 1 to 6. At 2024-02-23T12:55:19Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 6.
Successful	Launching a new EC2 instance: i-0b5581b2bdcf561e8	At 2024-02-23T12:55:16Z a monitor alarm TargetTracking-WebServerAutoScalingGroup-AlarmHigh-7e607729-4fc2-491f-87f1-603592f99d32 in state ALARM triggered policy Web Server Scaling Policy changing the desired capacity from 1 to 6. At 2024-02-23T12:55:19Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 6.
Successful	Terminating EC2 instance: i-099b4d32c351781af	At 2024-02-23T13:18:28Z a monitor alarm TargetTracking-WebServerAutoScalingGroup-AlarmLow-090cf4ca-180a-42f5-8c57-faf559279698 in state ALARM triggered policy Web Server Scaling Policy changing the desired capacity from 2 to 1. At 2024-02-23T13:18:40Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 2 to 1. At 2024-02-23T13:18:40Z instance i-099b4d32c351781af was selected for termination.
Successful	Terminating EC2 instance: i-067c61bc525c6597d	At 2024-02-23T13:17:28Z a monitor alarm TargetTracking-WebServerAutoScalingGroup-AlarmLow-090cf4ca-180a-42f5-8c57-faf559279698 in state ALARM triggered policy Web Server Scaling Policy changing the desired capacity from 3 to 2. At 2024-02-23T13:17:41Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 3 to 2. At 2024-02-23T13:17:41Z instance i-067c61bc525c6597d was selected for termination.
Successful	Terminating EC2 instance: i-0620f878fdad76f57	At 2024-02-23T13:16:34Z a monitor alarm TargetTracking-WebServerAutoScalingGroup-AlarmLow-ac956696-4f03-4e7b-8043-8f92d76f346e in state ALARM triggered policy Web Server Scaling Policy changing the desired capacity from 4 to 3. At 2024-02-23T13:16:43Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 4 to 3. At 2024-02-23T13:16:43Z instance i-0620f878fdad76f57 was selected for termination.
Successful	Terminating EC2 instance: i-04a8f07b1a2cf2293	At 2024-02-23T13:15:30Z a monitor alarm TargetTracking-WebServerAutoScalingGroup-AlarmLow-b3198908-b34c-4326-ad7d-55a01e32a935 in state ALARM triggered policy Web Server Scaling Policy changing the desired capacity from 5 to 4. At 2024-02-23T13:15:44Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 5 to 4. At 2024-02-23T13:15:44Z instance i-04a8f07b1a2cf2293 was selected for termination.

Figure 10. ETS Autoscaler Group Log

3.2 Cost Analysis

The Amazon Pricing Calculator will be used to provide a detailed estimate of the costs that are associated with the deployment of the web application on AWS. This allows a thorough evaluation of cost-effectiveness and budget alignment based on the requirements.

SERVICE	RESOURCE	MONTHLY COST (USD)	ANNUAL COST (USD)	DETAILS
Amazon EC2	t2.micro	4.23	50.76	An affordable option for the beginning phases of development and implementation.
Elastic Load Balancing	Application Load Balancer	16.51	198.12	Distribute incoming traffic across targets which will ensure high availability and reliability, calculation is based on average number of new connections per ALB (0.14), average connection duration (300 seconds), average number of request per second per ALB (1.39) and average number of rule (1)
Amazon RDS for MySQL	db.t2.micro	15.86	190.32	Cost efficient for ETS small database
Amazon CloudWatch	4 metrics 1 dashboard	1.20	14.40	Basic monitoring to ensure health and performance
TOTAL	-	**$37.8**	**$453.6**	

Assume that there is a surge in traffic for the web page, the overall cost is expected to increase because of these factors:

- **Amazon EC2:** More instances will be launched to support server load.
- **Elastic Load Balancing**: The costs will go up as traffic and connections increase, this requires more resources for effective distribution to various targets.
- **Amazon RDS for MySQL:** Additional costs will be necessary as more storage space is needed to handle increasing amounts of data.
- **Amazon CloudWatch:** It will be necessary to monitor a wider range of metrics to maintain operational resilience and performance effectiveness, leading to increased costs.

3.3 Reliability and Fault Tolerance

The system underwent a resilience assessment to evaluate its reaction to instance failure. This involved simulating an EC2 instance crash by intentionally terminating an operational EC2 instance. The occurrence was swiftly identified by Amazon CloudWatch.

During this test, the system maintained its performance and was fully accessible, demonstrating robust fault tolerance mechanisms that effectively mitigated any adverse effects of the simulated outage.

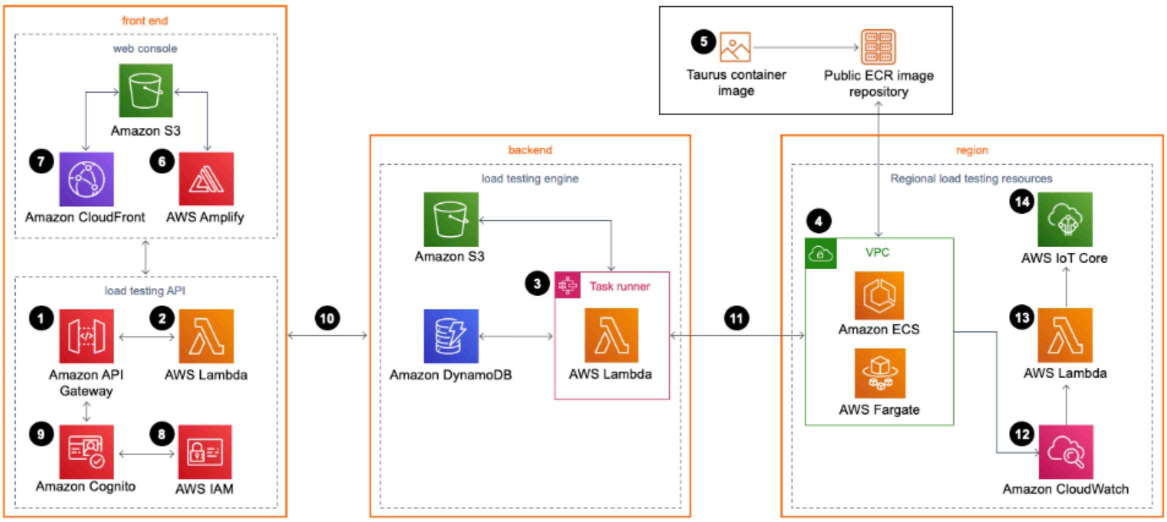

Figure 11. JMeter Reliability and Fault tolerance

MEMBER CONTRIBUTIONS

LEO EN QI VALERIE	RDS, Load Balancing and Auto Scaling, Report
OSAMA RASHEED KHAN	JMeter, Performance Evaluation and Testing, assist Backend integration, Report
SERI HANZALAH BTE HANIFFAH	Networking Setup and Security Measures, Cost Estimation, Report
TEO XUANTING	Cloud Architecture Design, Web Application, Report
TIAN YUE XIAO, BRYON	Cloud Architecture Design, Video Editing and Report