



MULTI-TIER
FEEDBACK
MANAGEMENT
SYSTEM

Dhivya Lakshmi S
Cloud Computing Technologies

Table of Contents

1. Project Overview	2
2. Phase 1: Architecture Highlights	2
○ Web Tier	
○ App Tier	
○ Database Tier	
○ Architecture Diagram	
3. Phase 2: Key AWS Services Used	4
4. Phase 3: Step-by-Step Implementation	4
○ Step 3.1: Create VPC & Networking	
○ Step 3.2: Security Groups	
○ Step 3.3: Database Tier Setup	
○ Step 3.4: IAM Role for App EC2	
○ Step 3.5: Route 53 Configuration	
5. Phase 4: Auto Scaling Configuration	14
○ AMI Setup	
○ Launch Template	
○ Auto Scaling Group (ASG)	
○ ALB Configuration	
○ Scaling Policy	
6. Phase 5: Multi-Tier & Auto Scaling Demonstration	17
○ Multi-Tier Proof	
○ Auto Scaling Proof	
7. Phase 6: Amazon Quick Sight Visualization	24
8. Phase 7: Key Learnings and Challenges Faced	25
9. Phase 8: Future Enhancements	26
10. Conclusion	27

PROJECT OVERVIEW

The **Multi-Tier Feedback Management System** is a cloud-native application built on **Amazon Web Services (AWS)**. It enables users to submit feedback through a web application, automatically performs **AI-powered sentiment analysis** using Amazon Comprehend, and visualizes insights using **Amazon Quick Sight dashboards**.

This project demonstrates how cloud technologies can be used to build **scalable, secure, and intelligent applications** that are ready for real-world traffic.

The system is designed in **three key layers**:

- **Web Tier:** User interface hosted on EC2 servers where feedback is submitted.
- **Database Tier:** Two RDS PostgreSQL databases – one for raw feedback and another for analyzed results.
- **Analysis & Visualization:** Feedback is processed by Amazon Comprehend, and insights are displayed in interactive Quick Sight dashboards.

To ensure **scalability and reliability**, the system uses **Application Load Balancers (ALB)** and **Auto Scaling Groups (ASG)**. This allows the application to dynamically adjust capacity based on incoming traffic.

PHASE 1: ARCHITECTURE HIGHLIGHTS

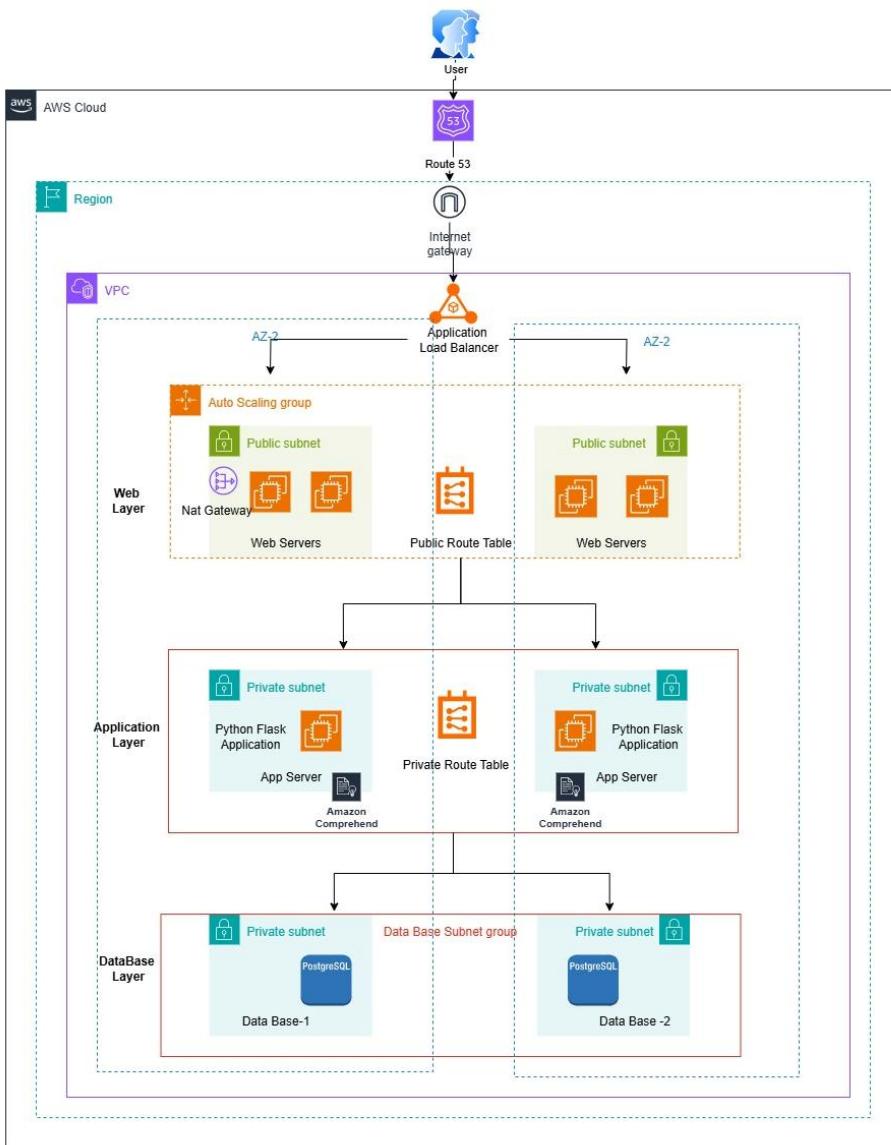
The system is built using a multi-tier design, which means different parts of the app are separated to improve security and performance.

Here's how the tiers work:

- **Web Tier:** Handles user requests through EC2 servers behind a public Application Load Balancer (ALB). This is where users interact with the app.
- **App Tier:** Runs the backend logic on EC2 instances behind an internal ALB. It processes feedback, calls Amazon Comprehend to analyze sentiment, and interacts with the databases.
- **Database Tier:** Contains two Amazon RDS databases (in private subnets) — one for storing raw feedback, and another for storing analyzed results.

Each tier is placed in its own subnet and controlled with Security Groups that only allow necessary traffic, keeping the system secure and organized.

Architecture Diagram:



PHASE 2. KEY AWS SERVICES USED

Web Tier	EC2, Auto Scaling, Internet-facing ALB
App Tier	EC2, Amazon Comprehend, Auto Scaling, Internal ALB
Database Tier	Amazon RDS (PostgreSQL)
Visualization	Amazon Quick Sight
Networking	VPC, Subnets, Route 53, Security Groups, IAM

PHASE 3. STEP-BY-STEP IMPLEMENTATION

Step 3.1: Create VPC & Networking

VPC (Virtual Private Cloud) is like your own private network in AWS, where you control how resources communicate and access the Internet.

- Public Subnets: These host resources that need to be accessed from the Internet, like your Web Tier EC2 servers.
- Private Subnets: These host resources that should not be directly accessible from the Internet, like your App and Database tiers, enhance security.
- Internet Gateway: A connection between your VPC and the Internet, allowing public subnets to send and receive traffic.
- Route Tables: Rules that direct network traffic within the VPC and to the Internet, ensuring communication between subnets and proper traffic flow.
- Create a custom VPC with public and private subnets across multiple Availability Zones.
- Public subnets for Web Tier, private subnets for App and DB tiers.
- Create and attach an Internet Gateway.
- Configure route tables for proper traffic flow.

vpc-0b610a65158f673a1 / Feedback-VPC

Details

VPC ID vpc-0b610a65158f673a1	State Available	Block Public Access Off	DNS hostnames Disabled
DNS resolution Enabled	Tenancy default	DHCP option set dopt-0fbf39ef873049795	Main route table rtb-0fb8c5906546b2be1
Main network ACL acl-003d6719d49d4c350	Default VPC No	IPv4 CIDR 10.0.0.16	IPv6 pool -
IPv6 CIDR -	Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 803871200222

Resource map

VPC Show details Your AWS virtual network	Subnets (4) Subnets within this VPC	Route tables (3) Route network traffic to resources	Network c Connections to
--	--	--	-----------------------------

subnet-06d42f2ee0362e5ef / Public Subnet-1

Details

Subnet ID subnet-06d42f2ee0362e5ef	Subnet ARN arn:aws:ec2:us-east-2:803871200222:subnet/subnet-06d42f2ee0362e5ef	State Available	Block Public Access Off
IPv4 CIDR 10.0.1.0/24	Available IPv4 addresses 250	IPv6 CIDR -	IPv6 CIDR association ID -
Availability Zone use2-az1 (us-east-2a)	VPC vpc-0b610a65158f673a1 Feedback-VPC	Route table rtb-0449f5cd7313735ab Public RT	Network ACL acl-003d6719d49d4c350
Default subnet No	Auto-assign public IPv4 address No	Auto-assign IPv6 address No	Auto-assign customer-owned IPv4 address No
Customer-owned IPv4 pool -	Outpost ID -	IPv4 CIDR reservations -	IPv6 CIDR reservations -
IPv6-only No	Hostname type IP name	Resource name DNS A record Disabled	Resource name DNS AAAA record Disabled
DNS64 Disabled	Owner 803871200222		

Flow logs

aws Search [Alt+S] United States (Ohio) Account ID: 8038-7120-0222 Dhivya-Lakshmi

VPC > Subnets subnet-04ae337fb277f72d

subnet-04ae337fb277f72d / Public Subnet-2

Details

Subnet ID	arn:aws:ec2:us-east-2:803871200222:subnet/subnet-04ae337fb277f72d	State	Available
IPv4 CIDR	10.0.2.0/24	IPv6 CIDR	-
Availability Zone	use2-az2 (us-east-2b)	Route table	rtb-0449f5cd7313735ab Public RT
Default subnet	No	Auto-assign IPv6 address	No
Customer-owned IPv4 pool	-	IPv4 CIDR reservations	-
IPv6-only	No	Resource name DNS A record	Disabled
DNS64	Disabled	Resource name DNS AAAA record	Disabled

Flow logs Route table Network ACL CIDR reservations Sharing Tags

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Search [Alt+S] United States (Ohio) Account ID: 8038-7120-0222 Dhivya-Lakshmi

VPC > Subnets subnet-044337e6dc359de6

subnet-044337e6dc359de6 / Private Subnet-1

Details

Subnet ID	arn:aws:ec2:us-east-2:803871200222:subnet/subnet-044337e6dc359de6	State	Available
IPv4 CIDR	10.0.3.0/24	IPv6 CIDR	-
Availability Zone	use2-az1 (us-east-2a)	Route table	rtb-0f6e51120c1284c14 Private RT
Default subnet	No	Auto-assign IPv6 address	No
Customer-owned IPv4 pool	-	IPv4 CIDR reservations	-
IPv6-only	No	Resource name DNS A record	Disabled
DNS64	Disabled	Resource name DNS AAAA record	Disabled

Flow logs Route table Network ACL CIDR reservations Sharing Tags

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Search [Alt+S] United States (Ohio) Account ID: 8038-7120-0222 Dhviya-Lakshmi

VPC > Subnets > subnet-01934913c2e9c67ac

subnet-01934913c2e9c67ac / Private Subnet-2

Details

Subnet ID	arn:aws:ec2:us-east-2:803871200222:subnet/subnet-01934913c2e9c67ac
IPv4 CIDR	10.0.4.0/24
Availability Zone	use2-az2 (us-east-2b)
Default subnet	No
Customer-owned IPv4 pool	-
IPv6-only	No
DNS64	Disabled
Subnet ARN	arn:aws:ec2:us-east-2:803871200222:subnet/subnet-01934913c2e9c67ac
State	Available
IPv6 CIDR	-
Route table	rtb-0f6e51120c1284c14 Private RT
Auto-assign IPv6 address	No
IPv4 CIDR reservations	-
Hostname type	IP name
Owner	803871200222
Resource name DNS A record	Disabled
Resource name DNS AAAA record	Disabled

Flow logs Route table Network ACL CIDR reservations Sharing Tags

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Search [Alt+S] United States (Ohio) Account ID: 8038-7120-0222 Dhviya-Lakshmi

VPC > Internet gateways > igw-061fd1a005a31aef1

igw-061fd1a005a31aef1 / Feedback-IGW

Details

Internet gateway ID	arn:aws:ec2:us-east-2:803871200222:internet-gateway/igw-061fd1a005a31aef1
State	Attached
VPC ID	arn:aws:ec2:us-east-2:803871200222:vpc-0b610a65158f673a1 Feedback-VPC
Owner	803871200222

Tags

Key	Value
Name	Feedback-IGW

Manage tags © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudShell Feedback

rtb-0449f5cd7313735ab / Public RT

Details

- Route table ID: rtb-0449f5cd7313735ab
- Main: No
- Owner ID: vpc-0b610a65158f673a1 | Feedback-VPC
- Explicit subnet associations: 2 subnets
- Edge associations: -

Routes (2)

Destination	Target	Status	Propagated	Route Origin
0.0.0.0/0	igw-061fd1a005a31aeef1	Active	No	Create Route
10.0.0.0/16	local	Active	No	Create Route Table

rtb-0f6e51120c1284c14 / Private RT

Details

- Route table ID: rtb-0f6e51120c1284c14
- Main: No
- Owner ID: vpc-0b610a65158f673a1 | Feedback-VPC
- Explicit subnet associations: 2 subnets
- Edge associations: -

Routes (1)

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	Create Route Table

Step 3.2: Security Groups

- **Web ALB SG** → Allow HTTP (80) from 0.0.0.0/0.
- **Web EC2 SG** → Allow HTTP (80) from Web ALB SG.
- **App EC2 SG** → Allow HTTP (80) from App ALB SG.
- **DB SG** → Allow MySQL (3306) from App EC2 SG.

aws Search [Alt+S] United States (Ohio) Account ID: 8038-7120-0222 Dhivya-Lakshmi

VPC > Security Groups sg-0e00604efea3f9981 - ALB SG

Subnets Route tables Internet gateways Egress-only internet gateways DHCP option sets Elastic IPs Managed prefix lists NAT gateways Peering connections Route servers New

▼ Security Network ACLs Security groups

▼ PrivateLink and Lattice Getting started Updated Endpoints Updated Endpoint services Service networks Updated Lattice services

CloudShell Feedback

sg-0e00604efea3f9981 - ALB SG

Details

Security group name	ALB SG	Security group ID	sg-0e00604efea3f9981	Description	VPC ID
Owner	803871200222	Inbound rules count	1 Permission entry	Outbound rules count	1 Permission entry
VPC ID: vpc-0b610a65158f673a1					

Inbound rules Outbound rules Sharing - new VPC associations - new Tags

Inbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-0b243db0cb8359e5d	IPv4	HTTP	TCP	80

Manage tags Edit inbound rules

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Search [Alt+S] United States (Ohio) Account ID: 8038-7120-0222 Dhivya-Lakshmi

VPC > Security Groups sg-0fe7d4141b380783f - EC2 SG

Subnets Route tables Internet gateways Egress-only internet gateways DHCP option sets Elastic IPs Managed prefix lists NAT gateways Peering connections Route servers New

▼ Security Network ACLs Security groups

▼ PrivateLink and Lattice Getting started Updated Endpoints Updated Endpoint services Service networks Updated Lattice services

CloudShell Feedback

sg-0fe7d4141b380783f - EC2 SG

Details

Security group name	EC2 SG	Security group ID	sg-0fe7d4141b380783f	Description	VPC ID
Owner	803871200222	Inbound rules count	2 Permission entries	Outbound rules count	1 Permission entry
VPC ID: vpc-0b610a65158f673a1					

Inbound rules Outbound rules Sharing - new VPC associations - new Tags

Inbound rules (2)

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-09ee1e27f5e08dd21	IPv4	SSH	TCP	22
-	sgr-06e4441b353d0c959	-	Custom TCP	TCP	5000

Manage tags Edit inbound rules

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

sg-01c5f7bef949a7628 - RDS SG

Details

Security group name RDS SG	Security group ID sg-01c5f7bef949a7628	Description RDS SG	VPC ID vpc-0b610a65158f673a1
Owner 803871200222	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Inbound rules | Outbound rules | Sharing - new | VPC associations - new | Tags

Inbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-0a00c9bae44ec2c1b	-	PostgreSQL	TCP	5432

Step 3.3: Database Tier Setup

- Create **DB1** (Raw feedback) and **DB2** (Analyzed feedback) in Amazon RDS.
- Place both in private subnets.
- Allow only App Tier EC2 SG access.

Aurora and RDS > Subnet groups > feedback-db-subnet-group

feedback-db-subnet-group

Subnet group details

VPC ID	vpc-0b610a65158f673a1
ARN	arn:aws:rds:us-east-2:803871200222:subgrp:feedback-db-subnet-group
Supported network types	IPv4
Description	Feedback-DB-Subnet-Group

Subnets (2)

Availability zone	Subnet name	Subnet ID	CIDR block
us-east-2b	Private Subnet-2	subnet-01934913c2e9c67ac	10.0.4.0/24
us-east-2a	Private Subnet-1	subnet-044337e6dc359de6	10.0.3.0/24

The screenshot shows the AWS RDS console with the database **feedback-db1**. The left sidebar is titled **Aurora and RDS** and includes sections for Dashboard, Databases, Query editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, and Events. The main content area has a title **feedback-db1** and a **Summary** section. The summary details include:

DB identifier	Status	Role	Engine	Recommendations
feedback-db1	Available	Instance	PostgreSQL	4 Informational
CPU	Class	Current activity	Region & AZ	
4.33%	db.t3.micro	0.00 sessions	us-east-2a	

Below the summary are tabs for Connectivity & security, Monitoring, Logs & events, Configuration, Zero-ETL integrations, Maintenance & backups, and Data. The Connectivity & security tab is selected, displaying information about the endpoint, networking, and security.

The screenshot shows the AWS RDS console with the database **feedback-db2**. The left sidebar is identical to the previous screen. The main content area has a title **feedback-db2** and a **Summary** section. The summary details include:

DB identifier	Status	Role	Engine	Recommendations
feedback-db2	Available	Instance	PostgreSQL	4 Informational
CPU	Class	Current activity	Region & AZ	
3.40%	db.t3.micro	0.00 sessions	us-east-2b	

Below the summary are tabs for Connectivity & security, Monitoring, Logs & events, Configuration, Zero-ETL integrations, Maintenance & backups, and Data. The Connectivity & security tab is selected, displaying information about the endpoint, networking, and security.

Step 3.4: Give your App EC2 permissions to call Amazon Comprehend so your sentiment analysis works.
Create an IAM Role for your App EC2

The screenshot shows the AWS IAM Roles page. On the left, a sidebar navigation includes 'Identity and Access Management (IAM)', 'Access management' (User groups, Users, Roles, Policies, Identity providers, Account settings, Root access management), 'Access reports' (Access Analyzer, Resource analysis, Unused access, Analyzer settings), and 'CloudShell' and 'Feedback' buttons. The main content area displays the 'EC2ComprehendRole' details. The 'Summary' tab shows the role was created on August 04, 2025, at 17:55 UTC-04:00. It has an ARN (arn:aws:iam::803871200222:role/EC2ComprehendRole) and an Instance profile ARN (arn:aws:iam::803871200222:instance-profile/EC2ComprehendRole). The 'Permissions' tab is selected, showing one policy attached: 'ComprehendFullAccess'. A 'Permissions boundary (not set)' section is present. At the bottom, there are links for 'Privacy', 'Terms', and 'Cookie preferences'.

Attach the role to your App EC2 instance

The screenshot shows the 'Modify IAM role' page for an EC2 instance. The instance ID is i-0ed30458b11f74b92 (app-ec2). The 'IAM role' dropdown is set to 'EC2ComprehendRole'. There are 'Create new IAM role' and 'Update IAM role' buttons. The bottom of the page includes 'CloudShell' and 'Feedback' buttons, and standard AWS footer links for 'Privacy', 'Terms', and 'Cookie preferences'.

Step 3.5: Route 53 configuration

Amazon Route 53 is a scalable and reliable Domain Name System (DNS) service that routes users to your application. It allows you to use a custom domain name instead of the default ALB URL.

- **Create a Hosted Zone** in Route 53 with your domain name.
- **Add an A record** that points your domain to your ALB (feedback-alb-962958100.us-east-2.elb.amazonaws.com).
- **Update your domain registrar** with the NS records from Route 53.

The screenshot shows the AWS Route 53 console. In the left sidebar, under the 'Domains' section, 'cloudzenhub.com' is selected. The main pane displays the 'Hosted zone details' for 'Public cloudzenhub.com'. Below this, the 'Records (3)' tab is selected, showing three entries:

Type	Value	TTL (s...)	Health ...	Evalu...
A	dualstack.feedback-alb-9629...	-	-	Yes
NS	ns-832.awsdns-40.net. ns-1977.awsdns-55.co.uk. ns-1169.awsdns-18.org. ns-21.awsdns-02.com.	172800	-	-
SOA	ns-832.awsdns-40.net. awsd...	900	-	-

PHASE 4. AUTO SCALING CONFIGURATION:

Auto Scaling automatically adjusts the number of EC2 instances in your application based on traffic and load. This ensures your system remains responsive during high demand and cost-effective during low usage.

- **AMI:** Prebuilt image with required software
 - Web AMI: Apache + website files

The screenshot shows the AWS EC2 AMIs page. On the left, there's a navigation sidebar with links like Instances, Images, Elastic Block Store, and Network & Security. The main content area is titled "Image summary for ami-0ea1e842482505b27". It displays various details about the AMI, such as its ID, name, owner account ID, root device name, status, creation date, and more. At the top right, there are buttons for "EC2 Image Builder", "Actions", and "Launch instance from AMI". The bottom of the page has tabs for "Permissions", "Storage", and "Tags".

- **Launch Template (per tier)**
 - AMI ID, instance type, SGs, IAM role
 - User data (start Apache / start Flask)

The screenshot shows the AWS EC2 Launch Templates page. The left sidebar includes links for Instances, Images, Elastic Block Store, and Network & Security. The main area is titled "feedback-launch-template (lt-02fd99a55460aadb3)". It shows the "Launch template details" section with fields for the launch template ID, name, default version, and owner. Below that is the "Launch template version details" section, which includes tabs for "Details", "Versions", and "Template tags". Under "Details", it lists the version (1), description, date created, and other instance-specific details like AMI ID, instance type, and security groups. At the top right, there are buttons for "Actions" and "Delete template version". The bottom of the page has tabs for "Actions", "Delete template", and "Delete template version".

ASG (Auto Scaling Group)

- Min: 1 • Desired: 1 • Max: 3
- Uses the Launch Template
- Attached to a Target Group

The screenshot shows the AWS EC2 Auto Scaling Groups console. The left sidebar navigation includes: Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups, Trust Stores), and Auto Scaling (Auto Scaling Groups). The main content area displays the 'feedback-asg' Auto Scaling Group. It shows the Capacity overview with Desired capacity: 1, Scaling limits (Min - Max): 1 - 3, Desired capacity type: Units (number of instances), and Status: -. Below this, the Date created is listed as Sat Aug 09 2025 15:28:15 GMT-0400 (Eastern Daylight Time). A navigation bar at the bottom includes Details, Integrations - new, Automatic scaling, Instance management, Instance refresh, Activity, and Monitoring.

• ALB (Application Load Balancer)

- Web: Public ALB → Web TG
- App: Internal ALB → App TG
- Health checks enabled

The screenshot shows the AWS EC2 Load Balancers console. The left sidebar navigation includes: Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups, Trust Stores), and Auto Scaling (Auto Scaling Groups). The main content area displays the 'feedback-alb' Application Load Balancer. It shows the Details section with Load balancer type: Application, Status: Active, Scheme: Internet-facing, VPC: vpc-0b610a6515bf673a1, Availability Zones: subnet-04ae337bfb277f72d (us-east-2b) and subnet-06d42f2ee0362e5ef (us-east-2a), and Load balancer IP address type: IPv4. The DNS name is info.feedback-alb-962958100.us-east-2.elb.amazonaws.com (A Record). Below this, theListeners and rules tab is selected, showing a table for Listener and Rule configuration. A note states: "A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules." A navigation bar at the bottom includes Actions, Details, Load balancer ARN, Network mapping, Resource map, Security, Monitoring, Integrations, Attributes, Capacity, and Tags.

Scaling Policy (Requests-based)

- Metric: RequestCountPerTarget (from ALB TG)

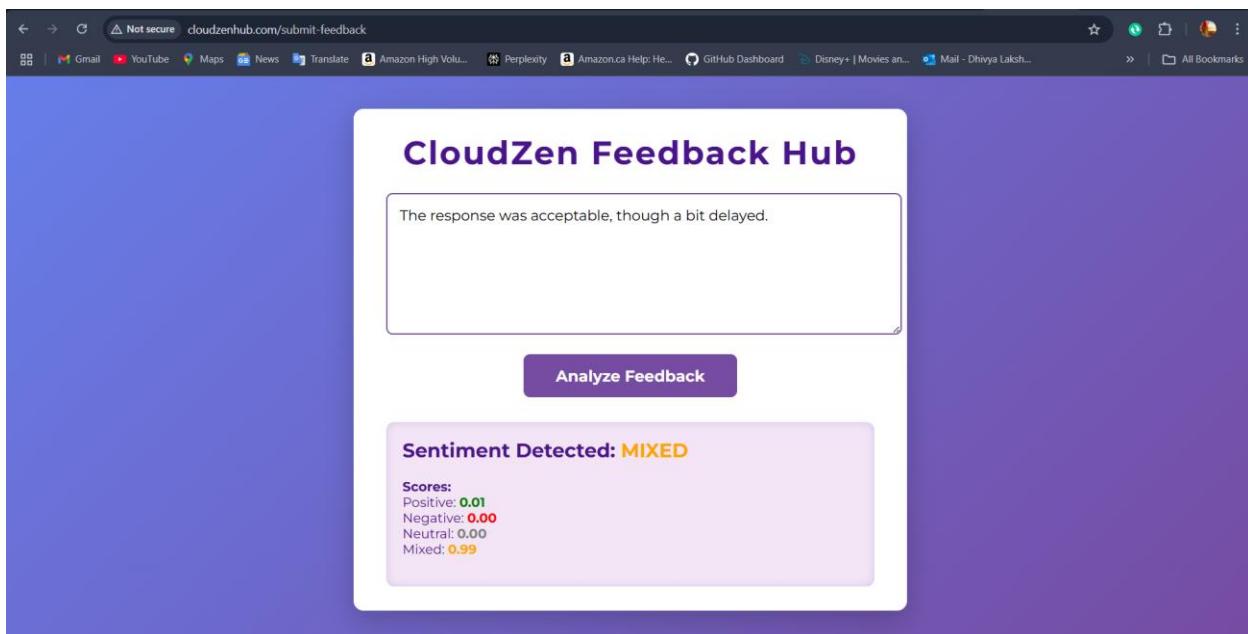
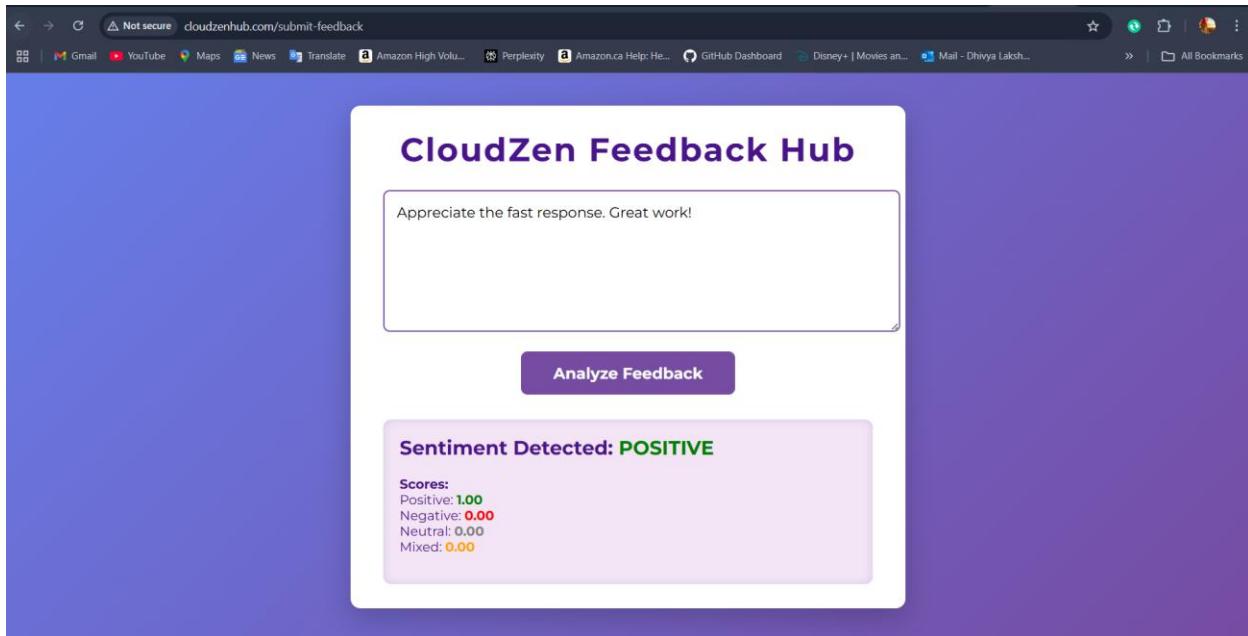
The screenshot shows the AWS Auto Scaling Groups page. On the left, the navigation menu includes EC2, Auto Scaling groups (selected), Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main content area displays a 'dynamic scaling policies' section for an 'ApplicationELB'. The policy type is 'Target tracking scaling', and it is 'Enabled'. The 'Execute policy when' condition is 'As required to maintain Application Load Balancer request count per target at 20'. The 'Take the action' is 'Add or remove capacity units as required'. The 'Instances need' condition is '120 seconds to warm up before including in metric'. The 'Scale in' condition is 'Enabled'. At the bottom, there is a 'Predictive scaling policies' section.

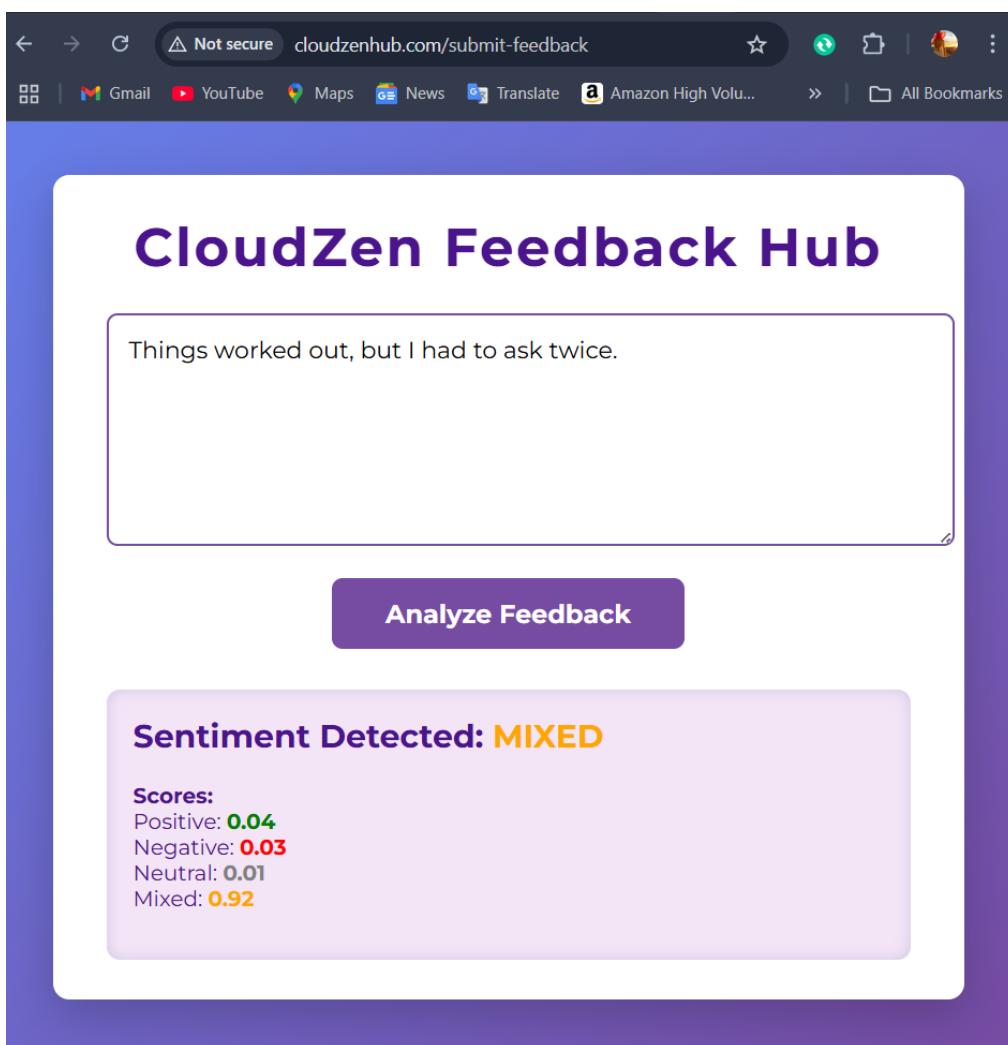
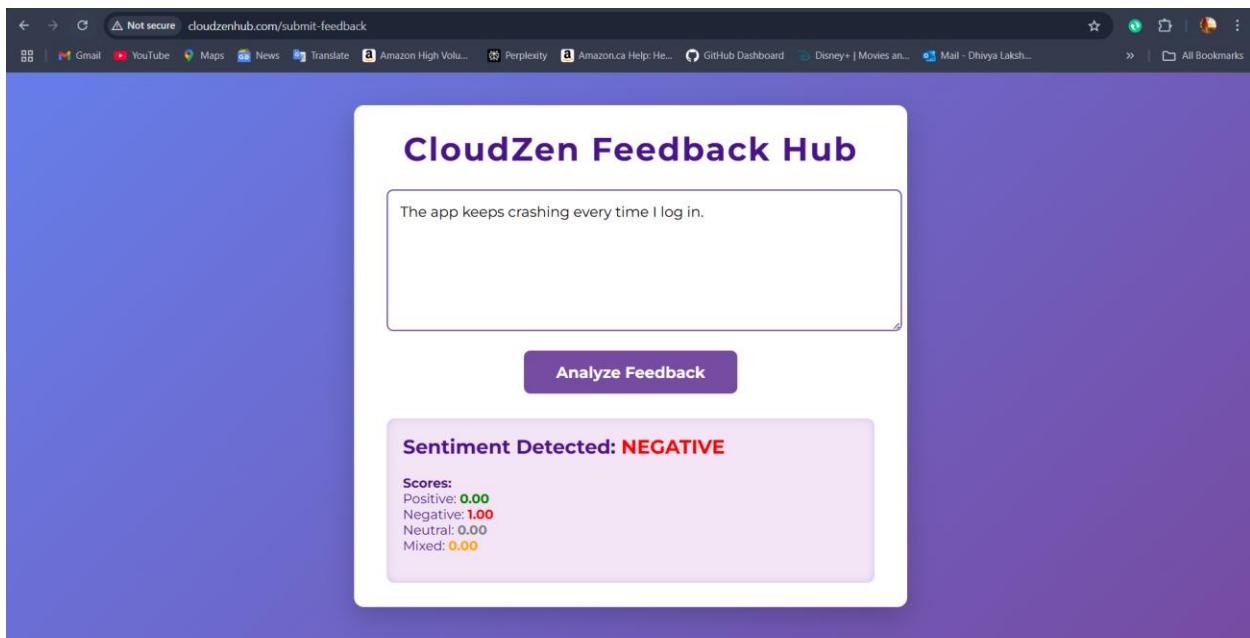
The screenshot shows the AWS Target Groups page. The navigation menu includes EC2 (selected), Target groups (selected), Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, and Elastic Block Store. The main content area shows the 'Feedback-TG' configuration. It lists the target type as 'Instance', protocol as 'HTTP: 5000', and protocol version as 'HTTP1'. The VPC is 'vpc-0b610a65158f673a1'. Below this, a table shows target status: 3 Total targets, 1 Healthy (green), 1 Unhealthy (red), 0 Unused, 0 Initial, and 1 Draining. A 'Distribution of targets by Availability Zone (AZ)' table is also present. At the bottom, tabs for Targets, Monitoring, Health checks, Attributes, and Tags are shown, along with a 'Registered targets (3)' section and buttons for Anomaly mitigation, Deregister, and Register targets.

PHASE 5. MULTI-TIER & AUTO SCALING ARE PROVEN IN THE DEMO

Multi-Tier Proof:

- User submits feedback from the browser.
- App Tier (Python Flask) stores raw text in DB1.
- Sends text to Amazon Comprehend for sentiment analysis.
- Stores analyzed result in DB2.





```

ec2-user@ip-10-0-2-41:~ 
Feedback_management=> ^C
feedback_management=> SELECT * FROM feedback_raw LIMIT 20;
 id | feedback_text | submitte
d_at
-----+-----+
 1 | This is good one | 2025-08-09 18:0
8:35.926178 | 2 | This is bad one | 2025-08-09 18:0
8:44.483967 | 3 | This is good one | 2025-08-09 19:3
2:40.050497 | 4 | Very friendly staff and quick resolution. | 2025-08-09 21:4
1:38.953853 | 5 | I love the new interface. Super intuitive! | 2025-08-09 21:4
1:49.716281 | 6 | I love the new interface. Super intuitive! | 2025-08-09 21:4
1:51.699331 | 7 | I got what I needed, but it took some time | 2025-08-09 21:4
1:57.58983 | 8 | Excellent customer support. Thank you! | 2025-08-09 21:4
2:06.172123 | 9 | Everything went smoothly. No complaints at all. | 2025-08-09 21:4
2:12.106044 | 10 | Everything went smoothly. No complaints at all. | 2025-08-09 21:4
2:14.520907 | 11 | Not bad, but could have been better | 2025-08-09 21:4
2:22.661248 | 12 | Super helpful! I got my issue resolved in minutes. | 2025-08-09 21:4
2:31.221873 | 13 | I had a wonderful experience with your team. | 2025-08-09 21:4
2:37.623145 | 14 | Appreciate the fast response. Great work! | 2025-08-09 21:4
3:48.675747 | 15 | The response was acceptable, though a bit delayed. | 2025-08-09 21:4
7:10.068798 | 16 | The app keeps crashing every time I log in. | 2025-08-09 21:4
7:48.675872 | 17 | The support representative was knowledgeable and polite. | 2025-08-09 21:5
7:46.433465 | 18 | Keep up the good work! | 2025-08-09 21:5
7:52.700955 | 19 | Loved the design and the usability. | 2025-08-09 21:5
8:00.433679

```

```

ec2-user@ip-10-0-2-41:~ 
Feedback_management=>
feedback_management=> SELECT * FROM feedback_analysis LIMIT 20;
 id | feedback_id | sentiment | analyzed_at | scores
-----+-----+
 1 | 1 | POSITIVE | {"Mixed": 0.0001586991274962201, "Neutral": 0.00016240045756101608, "Negative": 0.098636701965, "Positive": 0.9974344372749129} | 2025-08-09 18:36:015521
 2 | 2 | NEGATIVE | {"Mixed": 0.004838493186980486, "Neutral": 0.006489455234259367, "Negative": 0.98636701965, "Positive": 0.00230474357515735} | 2025-08-09 18:04:56439
 3 | 3 | NEUTRAL | {"Mixed": 0.0001586991274962201, "Neutral": 0.001824045756101608, "Negative": 0.0005828184657730562, "Positive": 0.9992814660023279} | 2025-08-09 18:32:405521
 4 | 4 | POSITIVE | {"Mixed": 0.0001273862924495034, "Neutral": 0.0065348646600352468, "Negative": 0.000582810103124631569, "Positive": 0.9992814660023279} | 2025-08-09 21:41:39.043085
 5 | 5 | POSITIVE | {"Mixed": 0.00009947031685442198, "Neutral": 0.0001775921555235982, "Negative": 0.00026170573619310744, "Positive": 0.9997861981319310744} | 2025-08-09 21:41:49.800057
 6 | 6 | NEUTRAL | {"Mixed": 0.00014855041627015453, "Neutral": 0.0006268011056818068, "Negative": 0.00026170573619310744, "Positive": 0.9997861981319310740} | 2025-08-09 21:41:51.744373
 7 | 7 | MIXED | {"Mixed": 0.9619448394775391, "Neutral": 0.005184868350625038, "Negative": 0.0177599180972576, "Positive": 0.012110341340303421} | 2025-08-09 21:41:57.693399
 8 | 8 | NEUTRAL | {"Mixed": 0.00014855041627015453, "Neutral": 0.0044448442176686823, "Negative": 0.00002611509080452379, "Positive": 0.9998192101123952} | 2025-08-09 21:42:06.258439
 9 | 9 | POSITIVE | {"Mixed": 0.000014855041627015453, "Neutral": 0.0006268011056818068, "Negative": 0.00004675102172768675, "Positive": 0.9993116855621338} | 2025-08-09 21:42:12.152156
10 | 10 | POSITIVE | {"Mixed": 0.000014855041627015453, "Neutral": 0.0006268011056818068, "Negative": 0.00004675102172768675, "Positive": 0.9993116855621338} | 2025-08-09 21:42:14.515846
11 | 11 | MIXED | {"Mixed": 0.9934212565422058, "Neutral": 0.0006371652125380933, "Negative": 0.004498714115470648, "Positive": 0.0014429463772103191} | 2025-08-09 21:42:22.743557
12 | 12 | POSITIVE | {"Mixed": 0.0002334579676631838, "Neutral": 0.00034360974677838385, "Negative": 0.00007447832467732951, "Positive": 0.9993484616279602} | 2025-08-09 21:42:31.303827
13 | 13 | NEUTRAL | {"Mixed": 0.00014855041627015453, "Neutral": 0.0006268011056818068, "Negative": 0.0000623541581574827, "Positive": 0.9996367692473887} | 2025-08-09 21:42:42.0530347173333, "Negative": 0.0000471629076, "Positive": 0.999798121452332} | 2025-08-09 21:43:48.759349
14 | 14 | POSITIVE | {"Mixed": 0.000018097094653057866, "Neutral": 0.0002072444766528904, "Negative": 0.00004475036490874365, "Positive": 0.999798121452332} | 2025-08-09 21:43:48.759349
15 | 15 | MIXED | {"Mixed": 0.989383285522461, "Neutral": 0.00014855041627015453, "Negative": 0.0002072444766528904, "Positive": 0.999798121452332} | 2025-08-09 21:43:51.157486
16 | 16 | NEGATIVE | {"Mixed": 0.000020317878806963563, "Neutral": 0.00006088128066039644, "Negative": 0.00004310367957, "Positive": 0.0001846998030923307} | 2025-08-09 21:47:48.78456
17 | 17 | POSITIVE | {"Mixed": 0.00017281246255151927, "Neutral": 0.00933252906799316, "Negative": 0.000080513115950801202, "Positive": 0.990391020534212503347173333} | 2025-08-09 21:47:48.78456
18 | 18 | POSITIVE | {"Mixed": 0.00065466028041213784, "Neutral": 0.00933252906799316, "Negative": 0.006824203883297741, "Positive": 0.9957948923110962} | 2025-08-09 21:57:52.791377
19 | 19 | POSITIVE | {"Mixed": 0.0000599955668207258, "Neutral": 0.00007237264071591198, "Negative": 0.00004230256308801472, "Positive": 0.999819383216858} | 2025-08-09 21:58:00.51436
20 | 20 | POSITIVE | {"Mixed": 0.0009436226177960634, "Neutral": 0.021783958435, "Negative": 0.00055546879548728, "Positive": 0.976863145828471} | 2025-08-09 21:58:06.458255
(20 rows)

```

Auto Scaling

- **Scale Out:** When the number of feedback requests increases, **CloudWatch monitors the RequestCountPerTarget metric**. If it exceeds the threshold, the **Auto Scaling Group (ASG)** automatically launches new EC2 instances to handle the traffic.
- **Scale In:** When requests decrease below the threshold, CloudWatch triggers the scale-in policy, and the **ASG terminates extra EC2 instances**, returning to minimum capacity.

```
ec2-user@ip-10-0-1-20:~$ ab -n 1000 -c 10 http://feedback-alb-962958100.us-east-2.elb.amazonaws.com/
Server Hostname: feedback-alb-962958100.us-east-2.elb.amazonaws.com
Server Port: 80

Document Path: /
Document Length: 3965 bytes

Concurrency Level: 10
Time taken for tests: 0.122 seconds
Complete requests: 100
Failed requests: 0
Total transferred: 414000 bytes
HTML transferred: 396500 bytes
Requests per second: 821.17 [#/sec] (mean)
Time per request: 12.178 [ms] (mean)
Time per request: 1.218 [ms] (mean, across all concurrent requests)
Transfer rate: 3319.98 [Kbytes/sec] received

Connection Times (ms)
              min     mean[+/-sd]   median     max
Connect:      0       0.3      0       3
Processing:   3      11.2      2.4     11      17
Waiting:     2      10.2      2.2     11      15
Total:        3      11.2      2.4     11      17

Percentage of the requests served within a certain time (ms)
  50%    11
  66%    12
  75%    13
  80%    13
  90%    15
  95%    15
  98%    16
  99%    17
100%    17 (longest request)

ec2-user@ip-10-0-1-20 ~]$ |
```

aws Search [Alt+S] United States (Ohio) Account ID: 8038-7120-0222 Dhivya-Lakshmi

EC2 > Target groups > Feedback-TG

Feedback-TG

Details

arn:aws:elasticloadbalancing:us-east-2:803871200222:targetgroup/Feedback-TG/3a162d81c66391d5	Target type: Instance	Protocol: Port: HTTP: 5000	Protocol version: HTTP1	VPC: vpc-0b610a65158f673a1	
IP address type: IPv4	Load balancer: feedback-alb				
3 Total targets	1 Healthy	1 Unhealthy	0 Unused	0 Initial	1 Draining
0 Anomalous					

Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.

Targets Monitoring Health checks Attributes Tags

Registered targets (3) info

Anomaly mitigation: Not applicable Deregister Register targets

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Search [Alt+S] United States (Ohio) Account ID: 8038-7120-0222 Dhivya-Lakshmi

EC2 > Auto Scaling groups > feedback-asg

feedback-asg

Filter activity history

Status	Description	Cause	Start time
Successful	Launching a new EC2 instance: i-0523b3453f78f6c39	At 2025-08-10T00:50:57Z an instance was launched in response to an unhealthy instance needing to be replaced.	2025 August 09, 08:50:59 PM -04:00
Connection draining in progress	Terminating EC2 instance: i-05380d8024f6a7fda - Waiting For ELB Connection Draining.	At 2025-08-10T00:50:57Z an instance was taken out of service in response to an ELB system health check failure.	2025 August 09, 08:50:57 PM -04:00
Successful	Launching a new EC2 instance: i-05380d8024f6a7fda	At 2025-08-10T00:44:51Z an instance was launched in response to an unhealthy instance needing to be replaced.	2025 August 09, 08:44:53 PM -04:00
Successful	Terminating EC2 instance: i-0be8e82544673e48b	At 2025-08-10T00:44:51Z an instance was taken out of service in response to an ELB system health check failure.	2025 August 09, 08:44:51 PM -04:00
	Launching a new EC2 instance: i-05380d8024f6a7fda	At 2025-08-10T00:38:57Z an instance was launched in response to an unhealthy instance needing to be replaced.	2025 August 09,

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS EC2 Auto Scaling Groups page. The left sidebar navigation includes EC2, Instances, Images, and Elastic Block Store. The main content area displays a table of instance events for the 'feedback-asg' group. The table columns are: Event Type, Instance ID, Description, and Time. The events listed are:

Event Type	Instance ID	Description	Time
Successful	Obe8e82544673e48b	replaced.	08:38:59 PM -04:00
Successful	Terminating EC2 instance: i-04cea6c97cebeae96	At 2025-08-10T00:38:57Z an instance was taken out of service in response to an ELB system health check failure.	2025 August 09, 08:38:57 PM -04:00
Successful	Launching a new EC2 instance: i-04cea6c97cebeae96	At 2025-08-10T00:33:03Z an instance was launched in response to an unhealthy instance needing to be replaced.	2025 August 09, 08:33:04 PM -04:00
Successful	Terminating EC2 instance: i-0a6284336e5c321ff	At 2025-08-10T00:33:02Z an instance was taken out of service in response to an ELB system health check failure.	2025 August 09, 08:33:03 PM -04:00
Successful	Launching a new EC2 instance: i-0a6284336e5c321ff	At 2025-08-10T00:26:58Z an instance was launched in response to an unhealthy instance needing to be replaced.	2025 August 09, 08:27:00 PM -04:00
Successful	Terminating EC2 instance: i-077b98ff85ace2189	At 2025-08-10T00:26:58Z an instance was taken out of service in response to an ELB system health check failure.	2025 August 09, 08:26:58 PM -04:00

Bottom navigation includes CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

The screenshot shows the AWS CloudWatch Alarms page. The left sidebar navigation includes CloudWatch, AI Operations, Logs, Metrics, Application Signals (APM), Network Monitoring, Insights, Settings, Telemetry config, and Getting Started. The main content area displays a table of alarms for the 'feedback-asg' group. The table columns are: Name, State, Last state update (UTC), Conditions, and Actions. The alarms listed are:

Name	State	Last state update (UTC)	Conditions	Actions
TargetTracking-feedback-asg-AlarmLow-cbbd72d2-a893-4e1d-8477-b426f1c250a0	OK	2025-08-13 07:31:24	RequestCountPerTarget < 14 for 15 datapoints within 15 minutes	Actions enabled
TargetTracking-feedback-asg-AlarmHigh-53844e93-cb90-484a-960f-69fab07c5000	OK	2025-08-13 07:25:34	RequestCountPerTarget > 20 for 3 datapoints within 3 minutes	Actions enabled

Bottom navigation includes CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

Screenshot of the AWS CloudWatch Alarms page showing the details of a metric alarm.

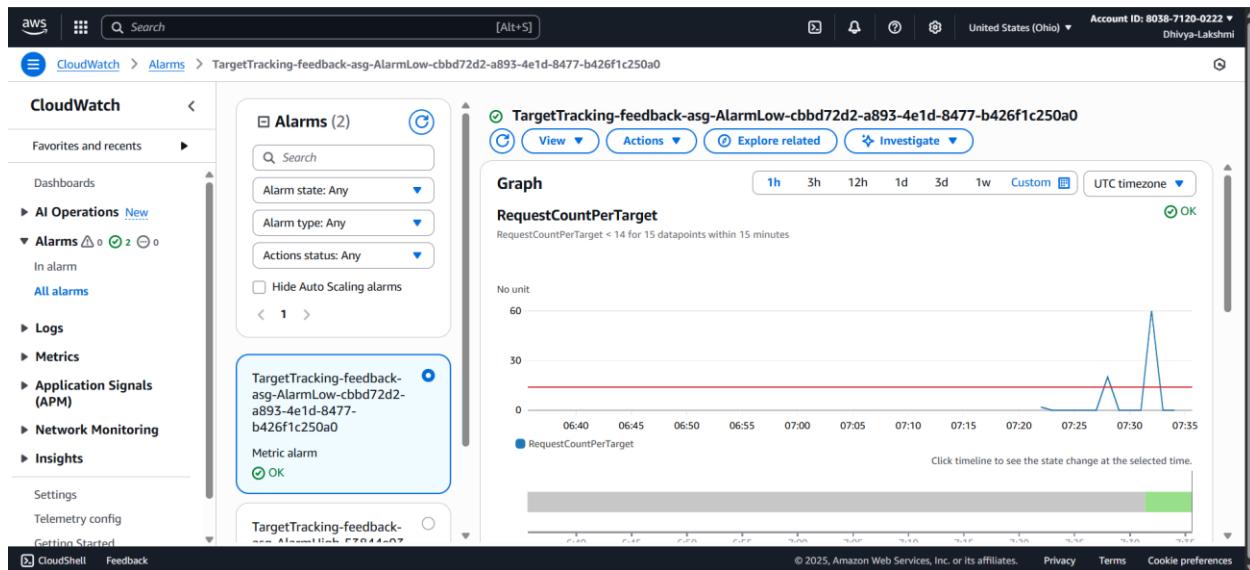
CloudWatch Alarms

TargetTracking-feedback-asg-AlarmLow-cbbd72d2-a893-4e1d-8477-b426f1c250a0

Details

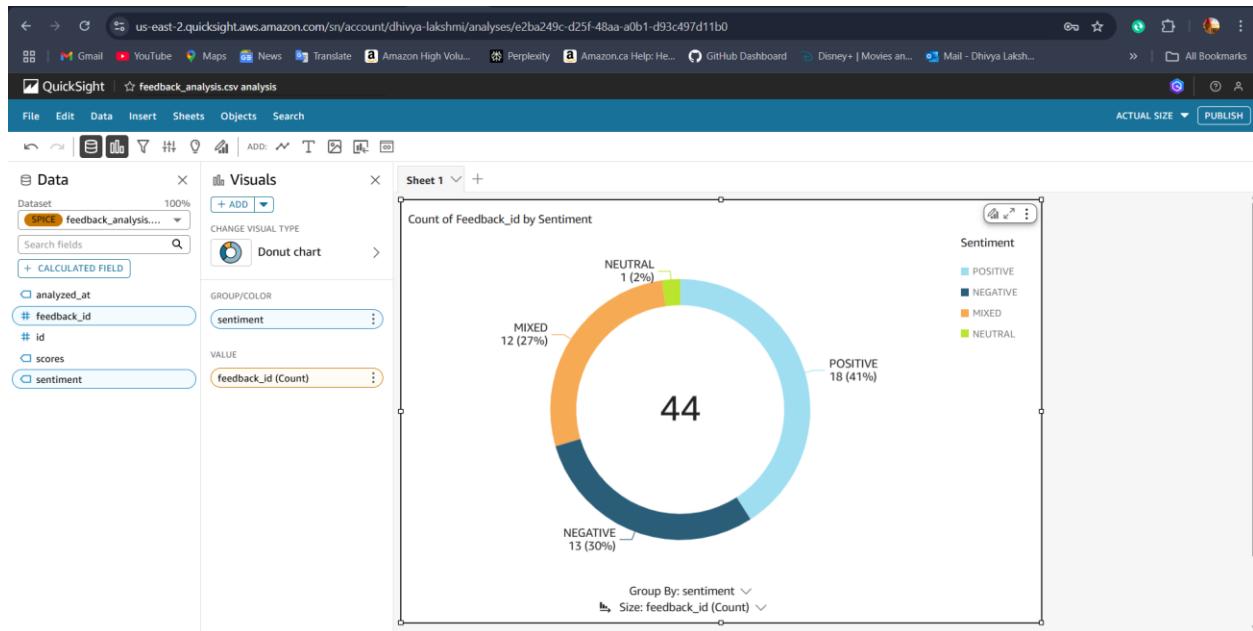
Name	TargetTracking-feedback-asg-AlarmLow-cbbd72d2-a893-4e1d-8477-b426f1c250a0	State	OK
Type	Metric alarm	Threshold	RequestCountPerTarget < 14 for 15 datapoints within 15 minutes
Description	DO NOT EDIT OR DELETE. For TargetTrackingScaling policy arn:aws:autoscaling:us-east-2:803871200222:scalingPolicy:1eb7d011-44cb-409d-aacc-4ecc4a7565dc:autoScalingGroup:feedback-asg:policyName/Application ELB.	Last state update	2025-08-13 07:31:24 (UTC)
Actions	Actions enabled	TargetGroup	targetgroup/Feedback-TG/5a162d81c66391d5
		Statistic	Sum
		Period	1 minute
		Unit	None

Datapoints to alarm: 15 out of 15. **Missing data treatment**: Treat missing data as missing. **Percentiles with low samples**: evaluate. **ARN**: arn:aws:cloudwatch:us-east-2:803871200222:alarm:TargetTracking-feedback-asg-AlarmLow-cbbd72d2-a893-4e1d-8477-b426f1c250a0



PHASE 6: AMAZON QUICKSIGHT

- Connect Quick Sight to DB2 (**Analyzed Feedback**).
- Create **interactive dashboards** showing:
 - Sentiment trends (positive, negative, neutral)
 - Feedback categories
 - Real-time insights for monitoring customer opinions



PHASE 7: KEY LEARNINGS

- **Multi-Tier Architecture:** Learned how to separate Web, App, and Database layers for better security, maintainability, and scalability.
- **AWS Services Integration:** Gained hands-on experience integrating EC2, RDS, ALB, Auto Scaling, Route 53, Amazon Comprehend, and Quick Sight.
- **Auto Scaling & Load Balancing:** Learned how to design systems that automatically scale based on traffic, ensuring high availability.
- **Security Best Practices:** Implemented Security Groups and IAM roles to restrict access and securely manage permissions.
- **Data Flow Understanding:** Learned end-to-end workflow from user input → database storage → AI analysis → visualization.

CHALLENGES FACED

- **ASG Scaling Issues:** Initially, instances did not scale because metrics from ALB were not triggering the policy correctly. Resolved by verifying CloudWatch metrics and cooldown settings.
- **ALB Request Distribution:** Understanding RequestCountPerTarget and ensuring the right load was sent to demonstrate scaling.
- **IAM Role Configuration:** Ensuring App EC2 had the correct permissions for Amazon Comprehend.
- **Database Connectivity:** Securing RDS in private subnets and allowing only App EC2 access.
- **Visualization Latency:** Ensuring QuickSight dashboards are updated near real-time with Comprehend-analyzed data.

PHASE 8: FUTURE ENHANCEMENTS

- **HTTPS Implementation:** Use AWS ACM to enable SSL and secure all traffic.
- **Advanced Analytics:** Implement more complex sentiment analysis, topic modeling, and trends over time.
- **Notification System:** Trigger alerts for negative feedback using SNS or Lambda.
- **Serverless Option:** Replace App EC2 with AWS Lambda to reduce costs and simplify scaling.
- **CI/CD Integration:** Automate deployments using Code Pipeline or GitHub Actions for faster updates.
- **Enhanced Dashboard:** Add filters, export options, and mobile-friendly views in Quick Sight.

REAL-TIME USAGE SCENARIOS

This system can be applied across industries to turn customer opinions into actionable insights:

- **E-Commerce Platforms**
Automatically analyze customer reviews and feedback on products to identify trending issues, improve product quality, and enhance customer satisfaction.
- **Hospitality & Travel**
Hotels, airlines, and travel agencies can collect guest feedback in real-time, measure satisfaction levels, and respond quickly to negative sentiment before it escalates.
- **Education & Training**
Universities and e-learning platforms can gather student feedback after lectures, courses, or training programs, helping educators refine teaching methods.
- **Healthcare**
Hospitals can analyze patient feedback to monitor service quality, detect dissatisfaction early, and improve patient care experiences.
- **Corporate HR & Employee Engagement**
Organizations can use this system internally to gather employee feedback, assess workplace sentiment, and make data-driven improvements to work culture.

CONCLUSION

The Multi-Tier Feedback Management System demonstrates how modern businesses can leverage cloud architecture, automation, and AI to process customer feedback at scale.

By combining EC2, RDS, ALB, Auto Scaling, Route 53, Comprehend, and QuickSight, this project showcases a secure, resilient, and intelligent system capable of handling variable workloads and providing actionable insights.

This hands-on project strengthened my expertise in:

- Cloud architecture design
- AWS deployment and automation
- AI integration for real-time insights
- Scalable and secure enterprise applications

This experience has built a strong foundation for architecting real-world enterprise-grade cloud applications.