WEBSITE LIVE VISITOR TRACKING SYSTEM

**Cloud Computing project Group 6**

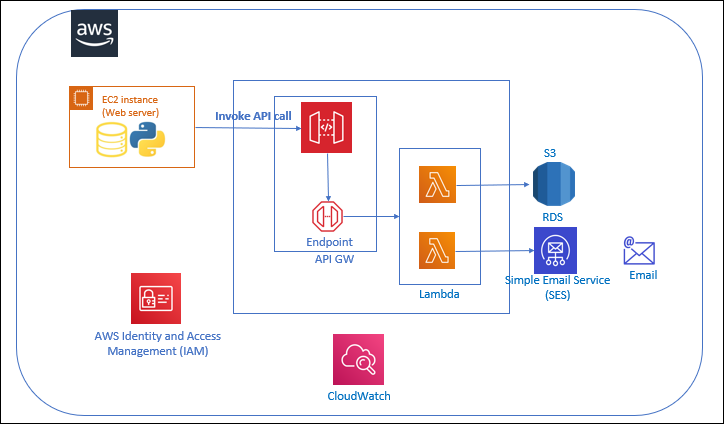
Vishnuvivek Balya(700747108)

Yerramilli Srivalli Vaishnavi(700741755)

Padarthi Pavani Sri(700741955)

Deepak Reddy Mailuru(700740099)

Architecture Diagram



**Website**

[***CLICK\_HERE\_FOR\_WEBSITE\_LINK***](https://logisticexpress-189637895473.s3.us-east-2.amazonaws.com/index.html)

Project Overview:

A live visitor tracking system for any website can be valuable in creating a cohesive and revenue-generating engine that accelerates sales growth. By collecting and analyzing data from the system, you can identify which content is attracting potential customers, how visitors are finding your website, and which of your marketing campaigns are generating the most attention. This information can then be used to diagnose any issues and make improvements to optimize your sales strategy.

Here we are tracking the number of users visiting the website and type of activity that is performed on hourly basis.

* The application has been developed using HTML, CSS & JS and it is being hosted on S3.
* The code is written in python
* The database used is MYSQL

To integrate with the backend system, the application will use an API that will be created and published using API Gateway. Requests to the API Gateway will then be forwarded to Lambda for processing. Lambda then has two functionalities   
  
1) one lambda function sends the live streaming data to store in S3  
2) Second lambda function creates a serverless platform to interact and store data in the Database (RDS)

Services Used:

**AWS API GW** – Through APIs, applications can gain access to data from backend services. It acts as an intermediate for these requests, allowing for secure and efficient communication between the application and backend systems.

**AWS Lambda** - AWS Lambda is a service that enables you to run code for various types of applications or backend services without the need to manage servers, making it a serverless and event-driven computing solution. It provides a flexible and scalable way to execute code based on events and requests, allowing you to focus on writing and deploying your code, rather than managing infrastructure.

**AWS Kinesis** : Amazon Kinesis is also a serverless service which captures the large/small scale data and then process it and store for future usage. This data can then be utilized for analysing and making business strategies.

**AWS CloudWatch** - CloudWatch is an Amazon Web Services (AWS) service that collects and stores monitoring and operational data in various forms such as logs, metrics, and events.

**AWS SES** - Amazon Simple Email Service (SES) is a versatile and cost-effective email service that provides the capability to send emails from within any application in a scalable and flexible manner.

**AWS IAM** - AWS Identity and Access Management (IAM) is a powerful web service that allows you to manage access to your AWS resources in a secure and scalable way.

**AWS RDS** - Amazon RDS is a comprehensive suite of managed services that simplifies the process of setting up, managing, and scaling databases in the cloud.

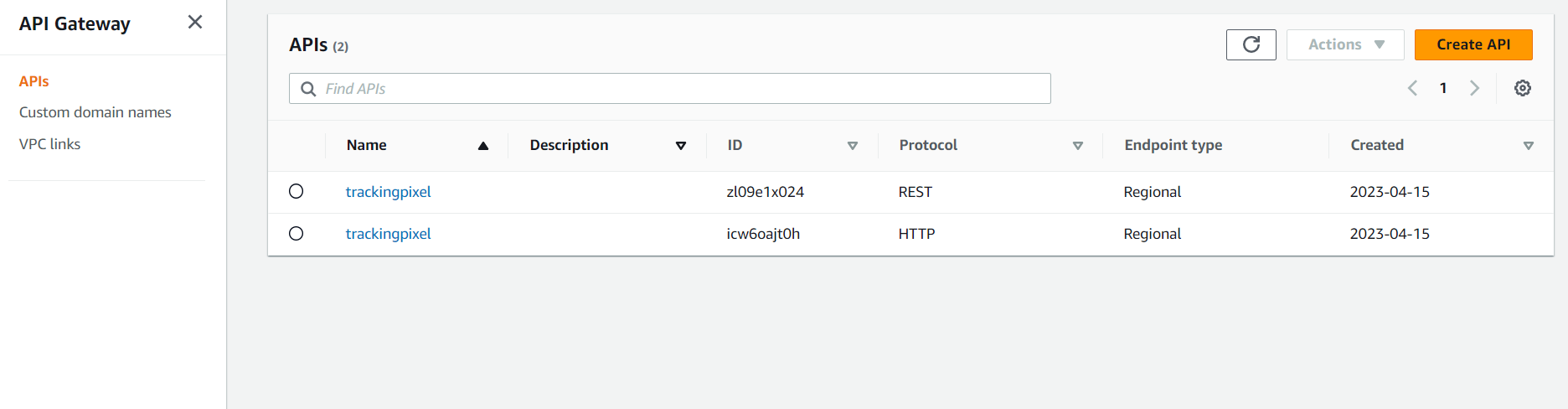
**AWS S3** - A object storage solution with remarkable scalability, dependability, data availability, security, and performance is Amazon Simple Storage Solution (Amazon S3).

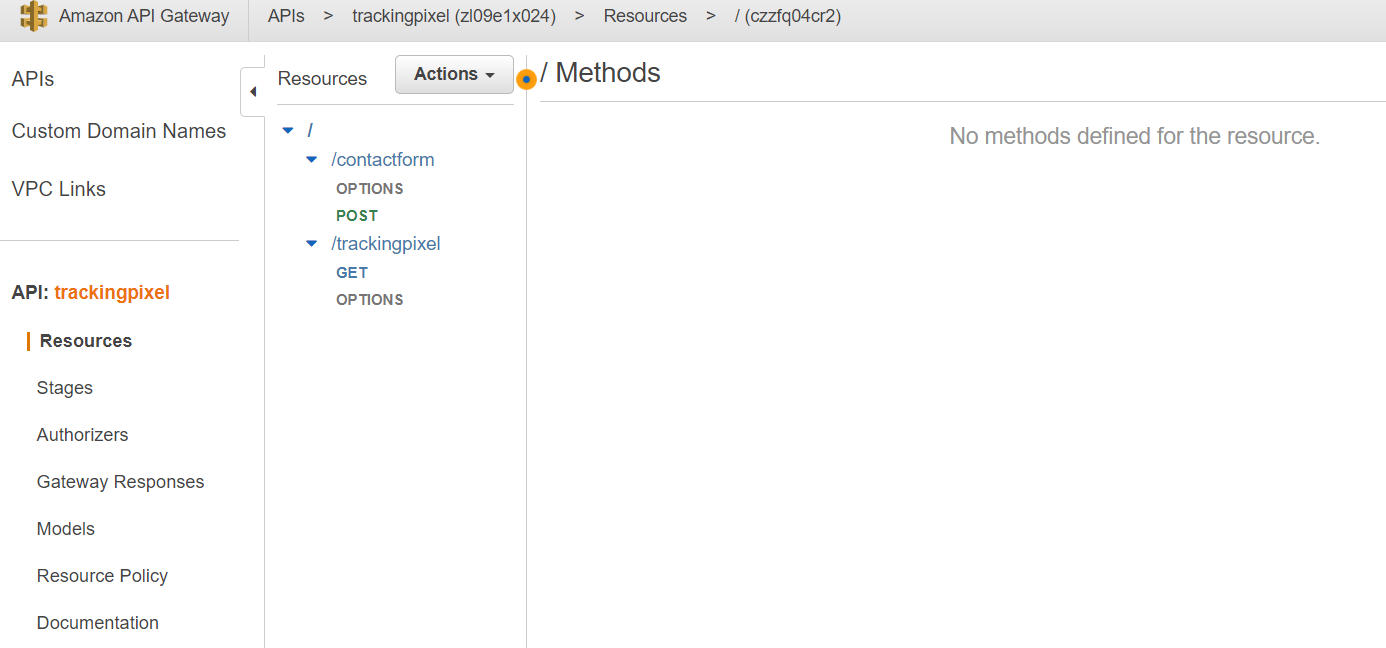
AWS API GW:

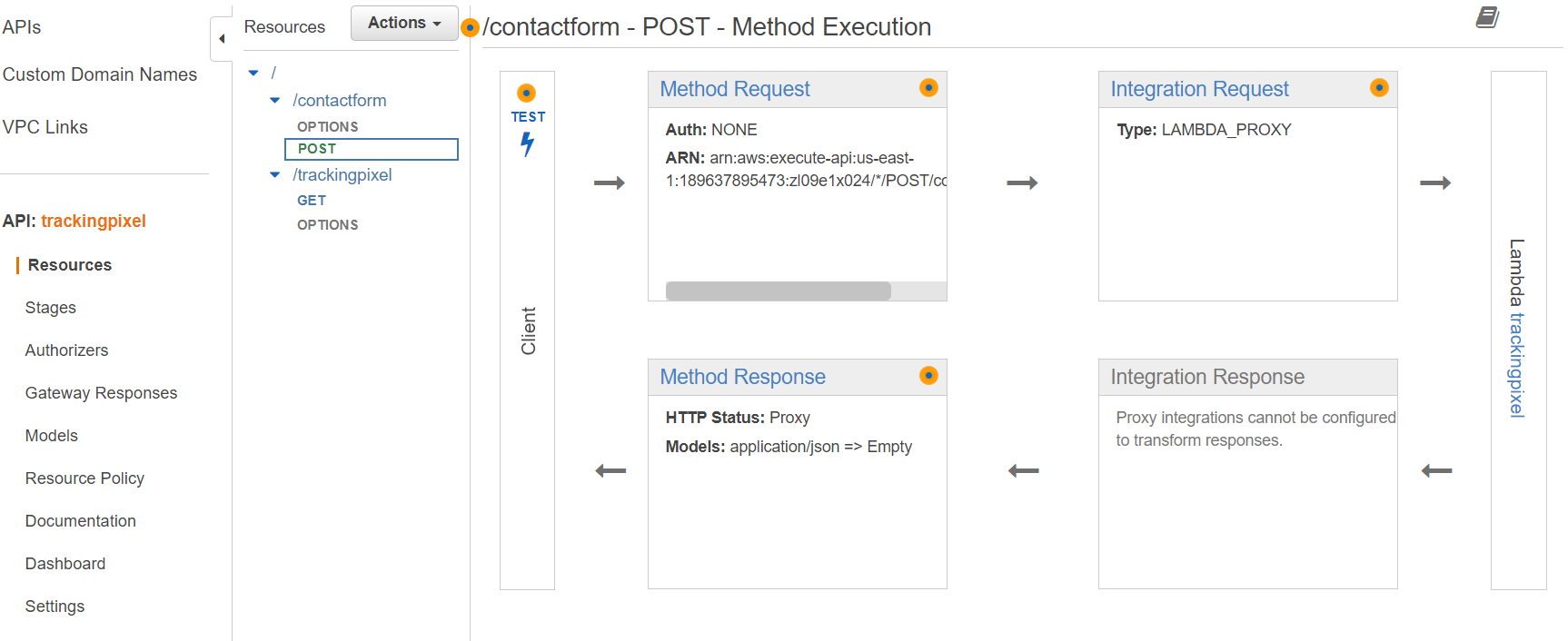
With Amazon API Gateway, managing APIs is a breeze as it is a managed service that simplifies their creation, publishing, maintenance, monitoring, and security, regardless of their scale. APIs act as the primary entry point for applications to access backend services such as data, business logic, and functionality. You may build real-time two-way communication apps using WebSocket and RESTful APIs with the aid of API Gateway. It is compatible with both containerized and serverless workloads and web applications.

API Gateway can provide traffic management, Cross-Origin Resource Sharing support, authorization, and access control, limiting the number of API requests a user can make in a certain period i.e., throttling, monitoring, and API version management, among other duties required in accepting and processing many API calls concurrently. With API Gateway, there are no setup fees or minimum charges; instead, you only pay for the API calls you make and the data that is sent out. Moreover, API Gateway's tiered pricing model helps you to cut down the costs based on your API usage scales.

In our website live tracking system, we are hosting a static serverless webpage over S3. There are two web pages on our website i.e., the Home and Contact Us pages. We are invoking the REST and HTTP API gateways through these pages, which are integrated with the lambda functions to collect the data.







AWS Lambda:

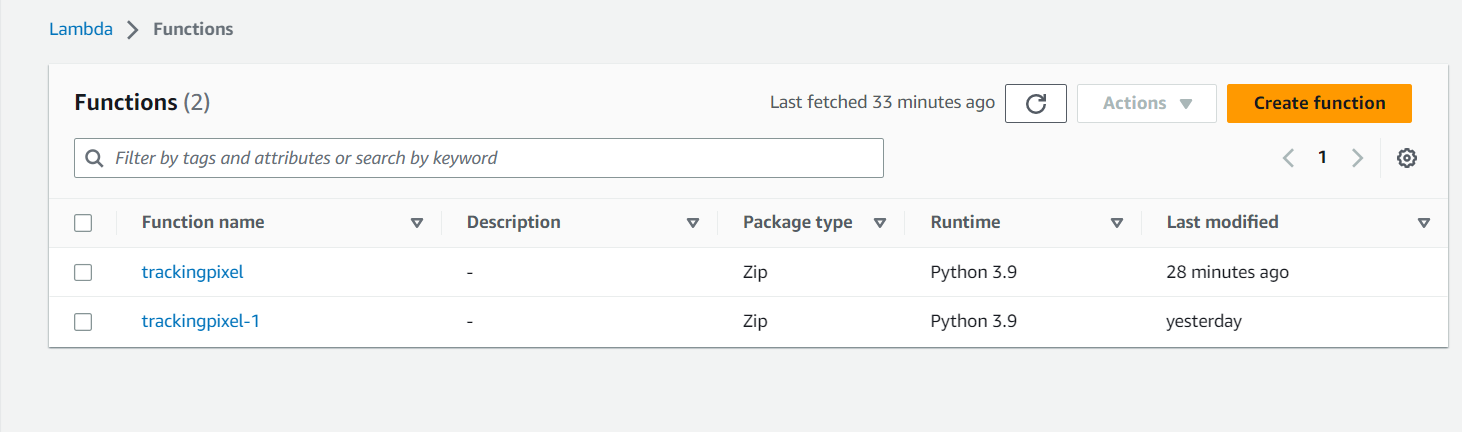
AWS Lambda is a compute service that operates on an event-driven, serverless architecture, enabling you to execute code for a wide range of applications or backend services without the need for server provisioning or management. You can execute code without the hassle of provisioning or managing infrastructure by simply writing and uploading your code as a .zip file or container image using AWS Lambda. It automatically handles code execution requests at any scale, from a few events per day to hundreds of thousands per second.

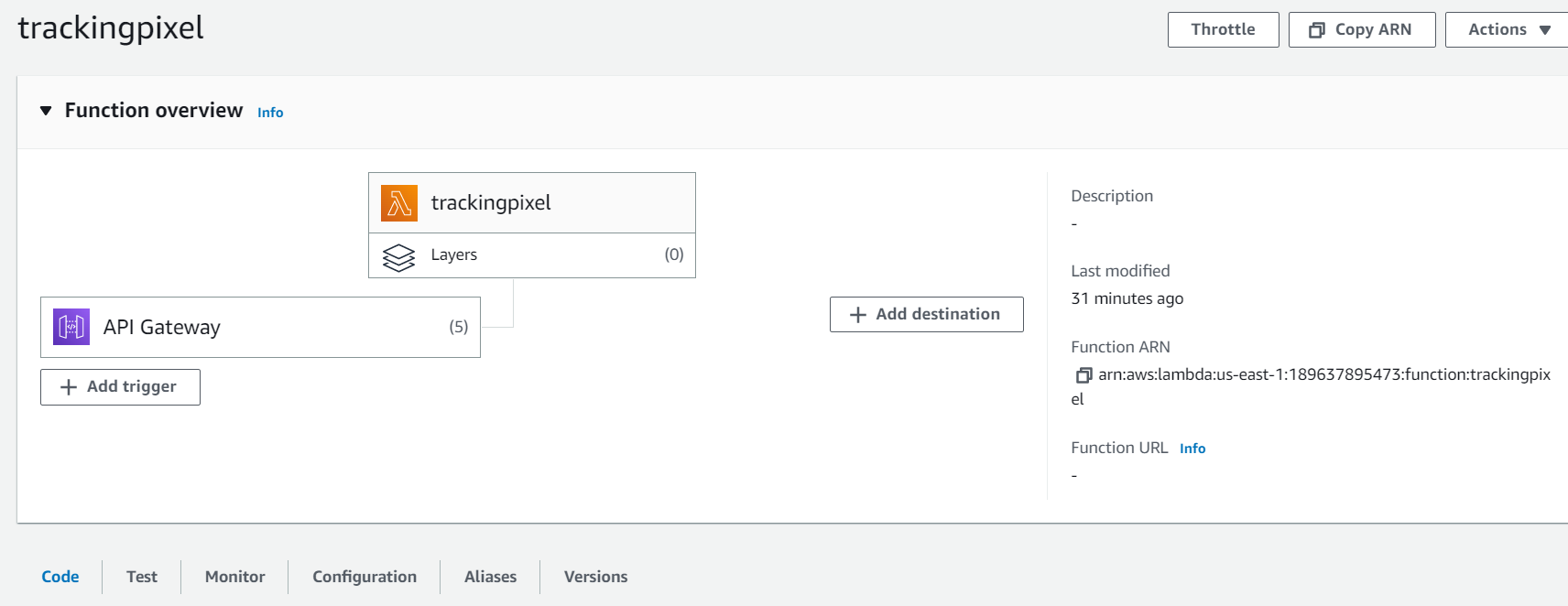
With AWS Lambda, you can spend the minimal by paying for the exact compute time you consume, charged per millisecond, instead of paying for a hefty amount on infrastructure for provisioning during peak capacity. You can also optimize the performance and execution time of your code by selecting the appropriate function memory size. To handle high demand in mere milliseconds, you can use Provisioned Concurrency.

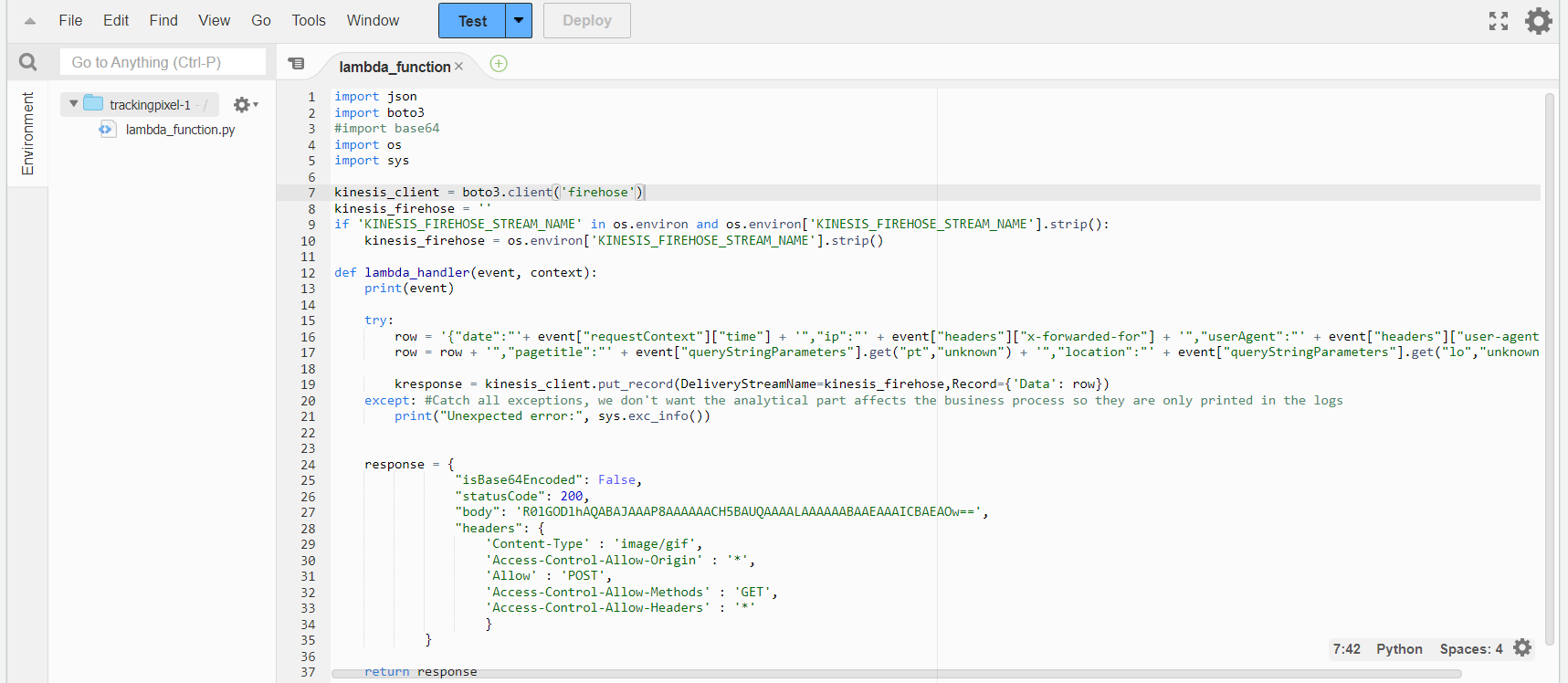
In our project, we have created two lambda functions i.e., trackingpixel and trackingpixel-1. In AWS Lambda, code runs in response to some events i.e., HTTP/REST requests from the API gateway these events trigger the lambda function to execute the code.

Whenever the user visits the home page, this event invokes the HTTP API gateway which is integrated with the lambda function(pixel1) with the help of kinesis transfers the client requests i.e., streaming data, and dumps the data into the S3 bucket.

Correspondingly, Whenever the user contacts the client through the Contact Us page to enquire/resolve their queries, this event invokes the REST API gateway which is integrated with the other lambda functions (pixel), It transfers the client requests i.e., the data to store in the SQL database i.e., AWS RDS





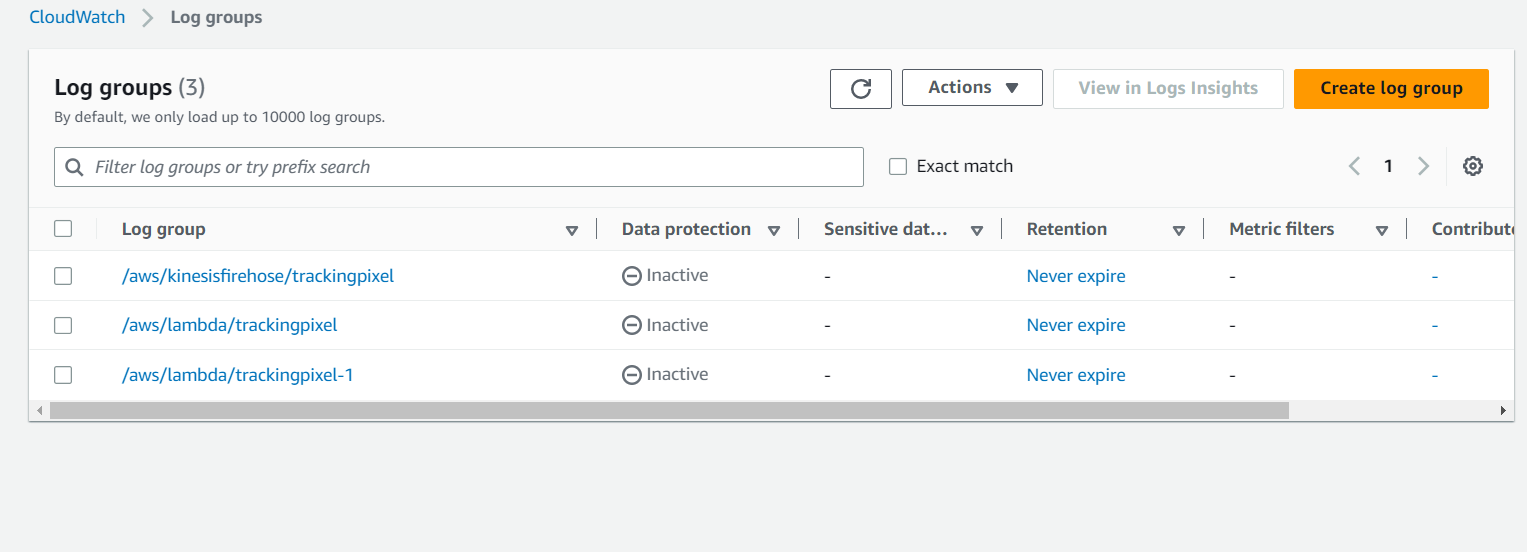


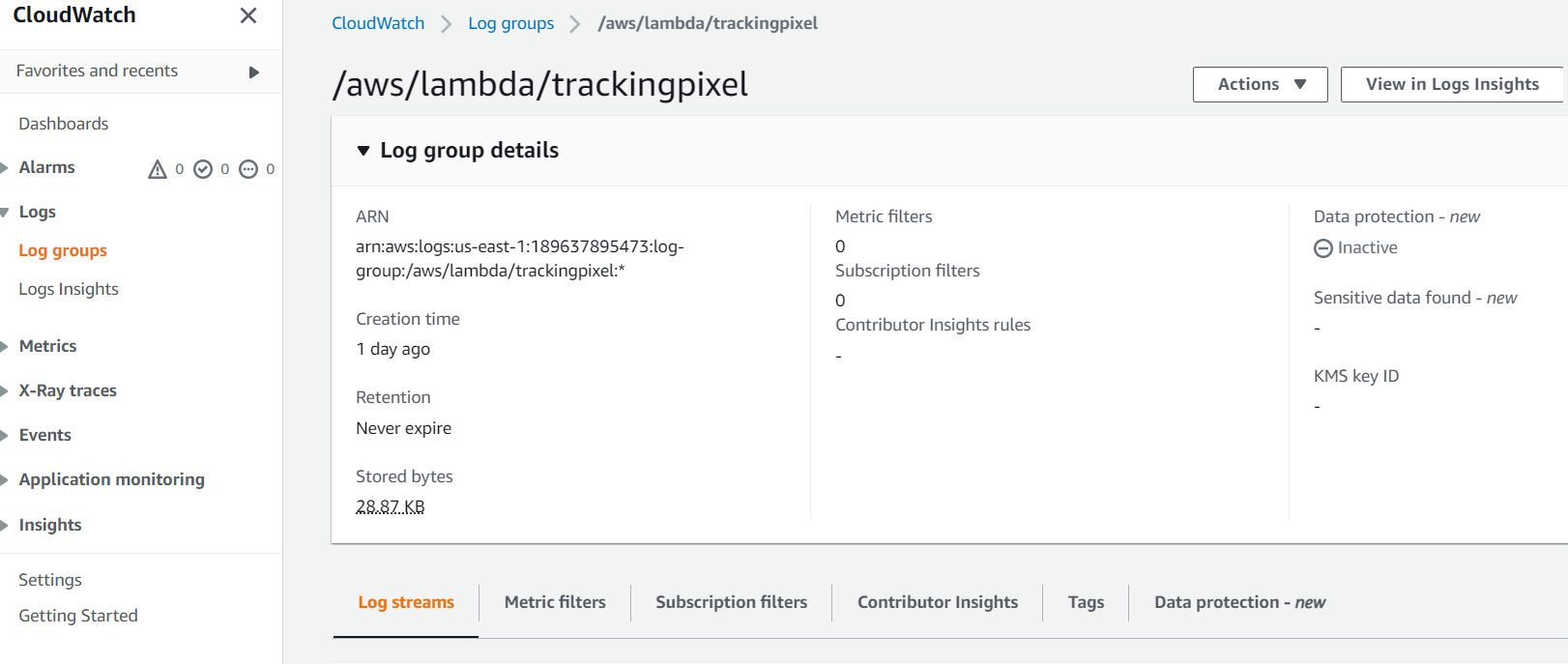
AWS CloudWatch:

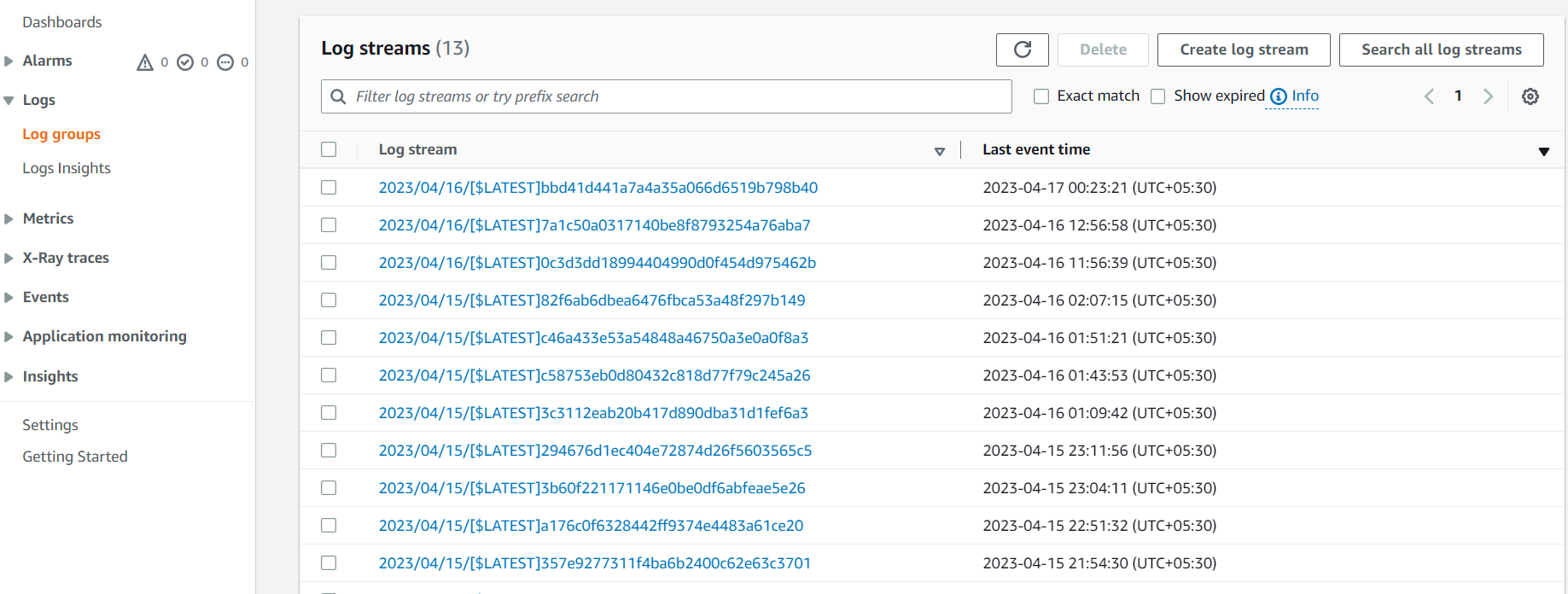
Amazon CloudWatch is one of the unique monitoring services designed by the AWS cloud team to cater to the needs of techies in the IT sector, DevOps engineers, and product owners. CloudWatch provides you with the data and insights which help you to keep track of apps, respond to system-wide performance variations, and improve resource consumption. With CloudWatch, you can identify abnormal behavior in your environments, create alarms, compare and contrast logs and metrics, take automated actions, address problems, and gain insights to maintain the seamless operation of your applications.

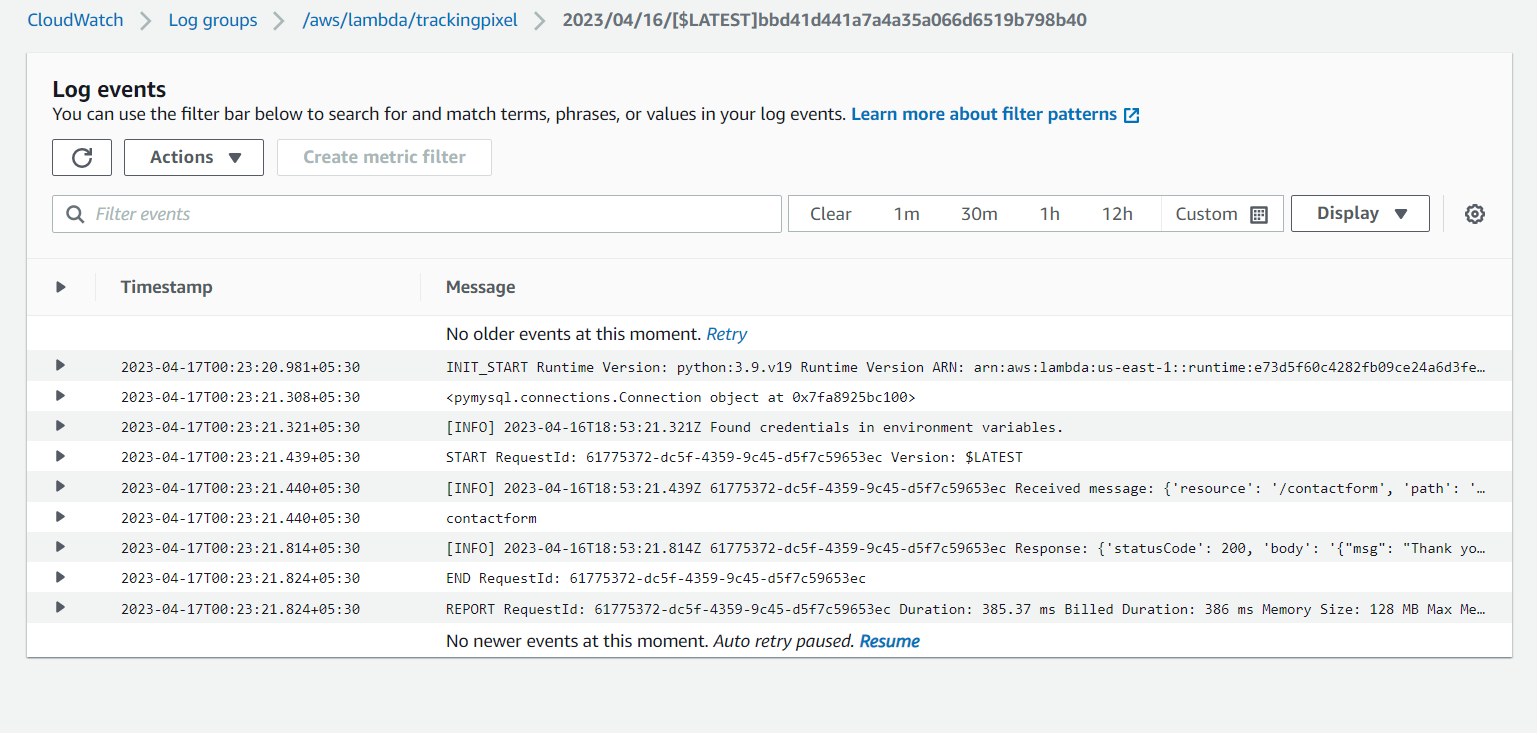
CloudWatch gathers operational and monitoring data, such as logs, metrics, and events, providing you with a comprehensive overview of your operational health. With this, you gain full visibility of your AWS resources, applications, and services, both on-premises and running on AWS.

In this project, AWS CloudWatch is used in order to detect the anomalies in the website if it persists or to extract the data insights from the user to track, improve and smooth the functioning of the website. We have collected the user information in the form of logs in log groups.







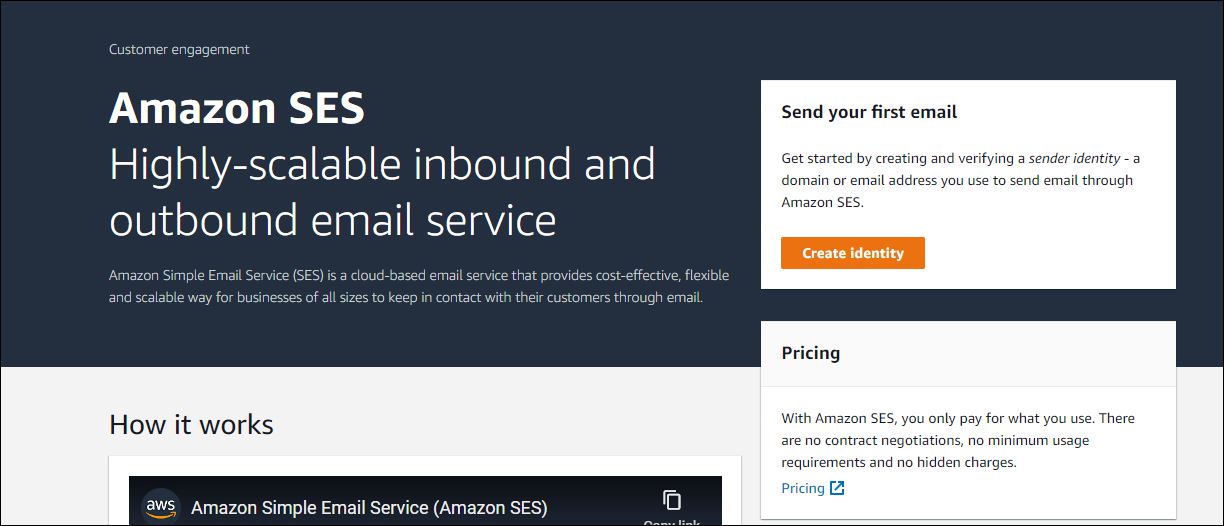


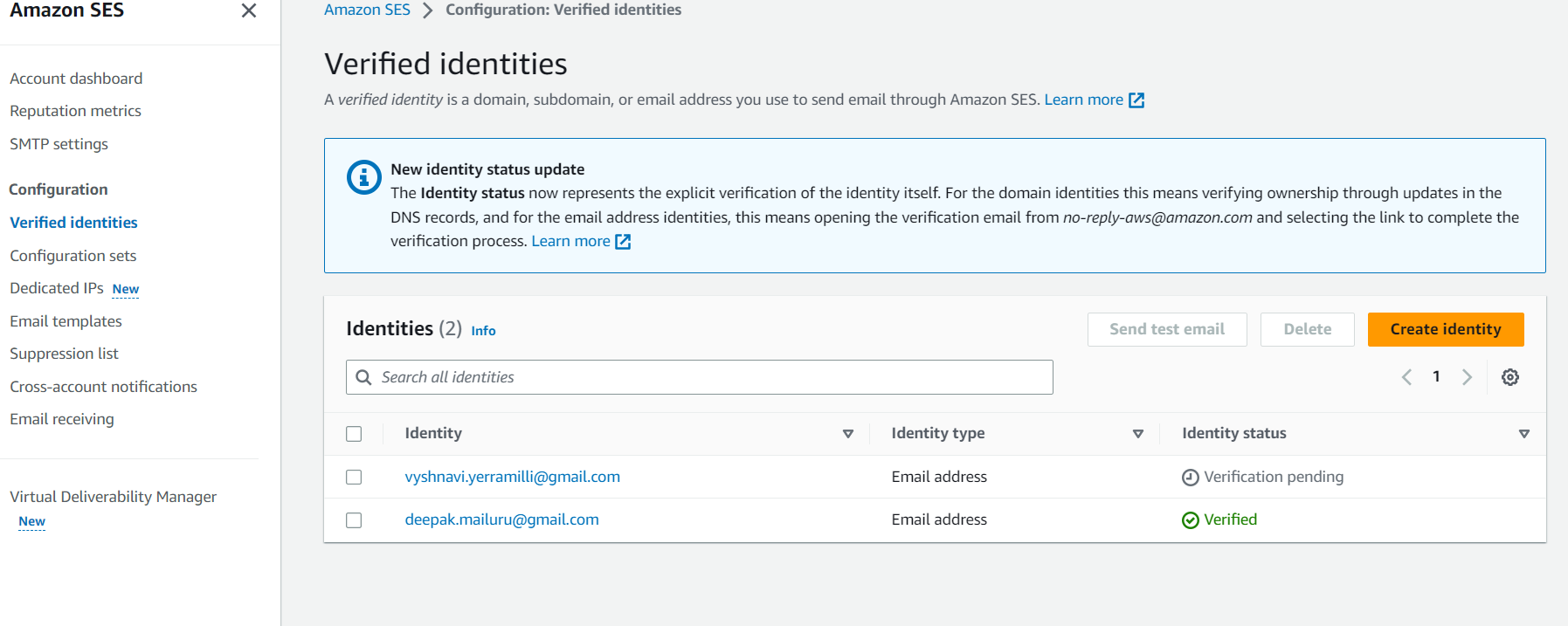
AWS SES:

Amazon Simple Email Service (SES) is an email service that is both cost-effective and flexible, allowing developers to send emails from any application in a scalable manner. You can easily set up Amazon SES to cater to various email use cases, such as Business-related, Advertising, or mass email communications.

Amazon SES's IP deployment flexibility and email authorization options improve deliverability and safeguard the sender's reputation while providing analytics to assess the impact of each email. With Amazon SES, you can send emails safely, worldwide, and at any scale.

In our website live tracking project, AWS SES plays a pivotal role. Whenever a user interacts with the client through the contact us page, In AWS SES configuration, we have given the admin privileges i.e., verified identity to the client mail id. No matter when a user tries to interact with the client, the client receives the notification, and he may address the issue instantly which helps in the smooth functioning of the company.



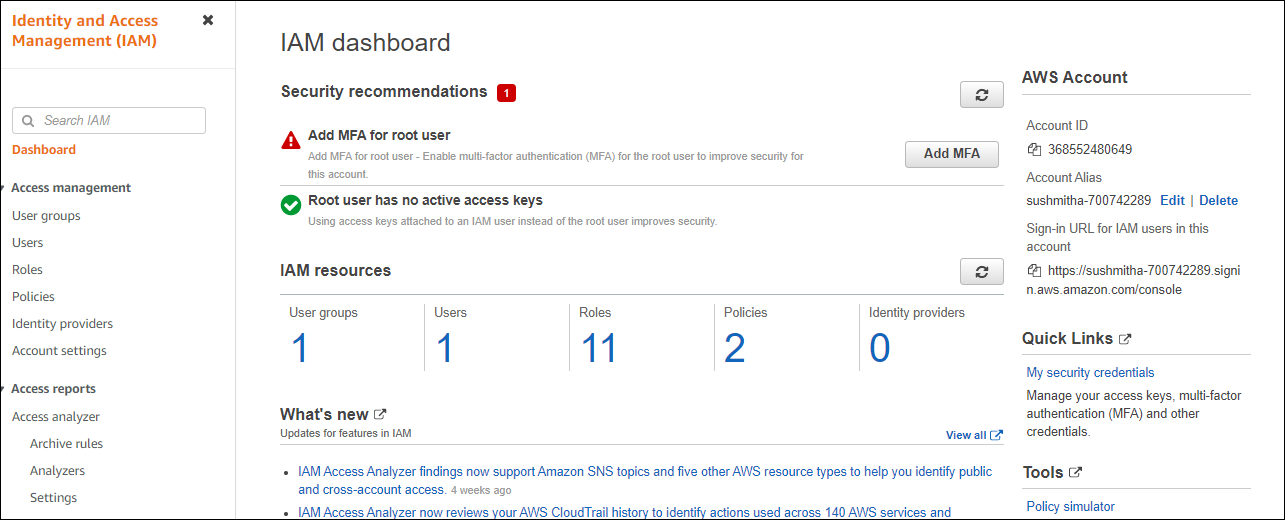


AWS IAM:

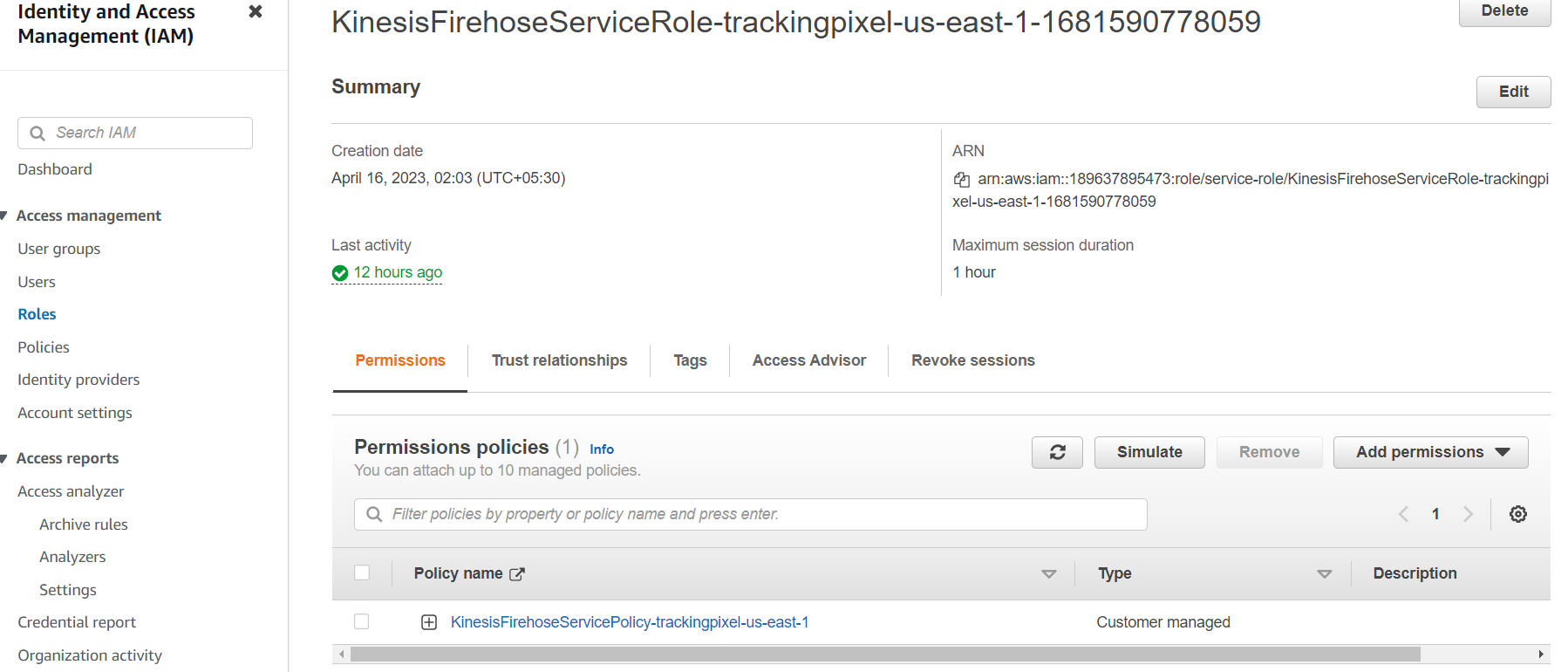
AWS Identity and Access Management (IAM) is a service provided by AWS that enables secure management of access to resources on their platform.

There are six key elements included in the IAM sequence, which are:

* An entity that has the capability to perform actions on an AWS resource is referred to as a principal. This can include a user, a role, or an application.
* Authentication refers to the process of verifying the identity of a principal attempting to access an AWS resource. The principal is required to provide their credentials or necessary keys to authenticate themselves.
* Request: When a principal wants to perform an action on a resource in AWS, they send a request to the platform indicating the specific action they want to take and which resource they want to perform it on.
* Authorization: In AWS, resources are denied access by default. IAM only grants authorization to a request if all components of the request comply with a matching policy. Once the desired request has been authenticated and authorized, AWS approves the action.
* Actions in AWS are used to perform various operations on a resource, such as viewing, creating, editing, or deleting it.
* Resources: There is a set of actions that can be performed on an AWS resource that is associated with your account.
* In this project, it is a tedious process to give access to the user to each individual resource instead of this task, We have updated the AWS policies and provided access to the user based on the request.





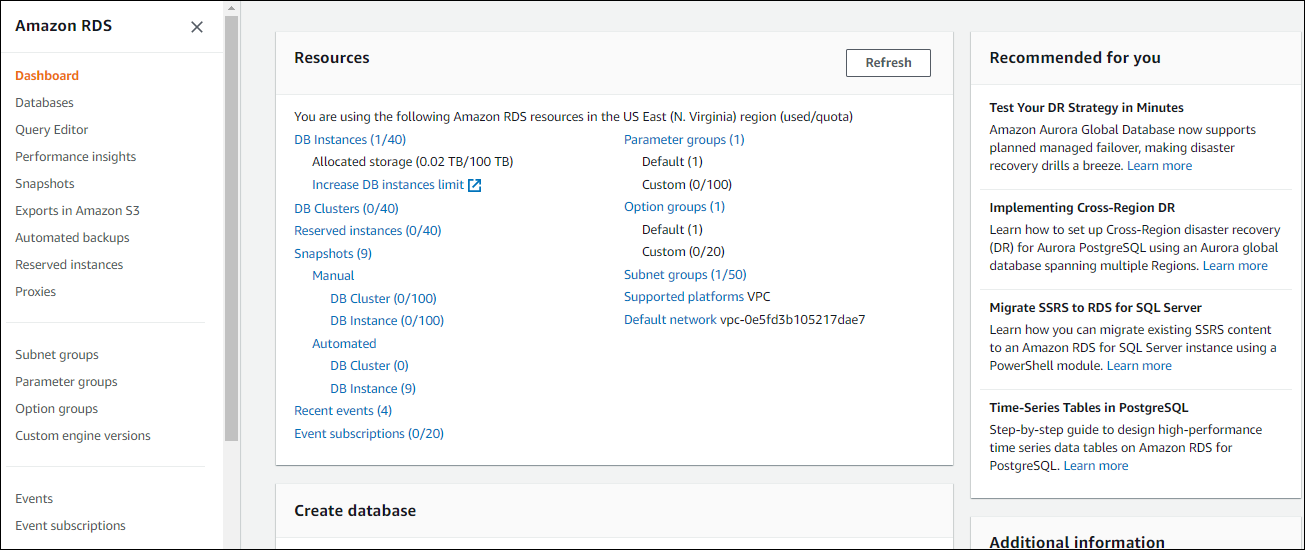


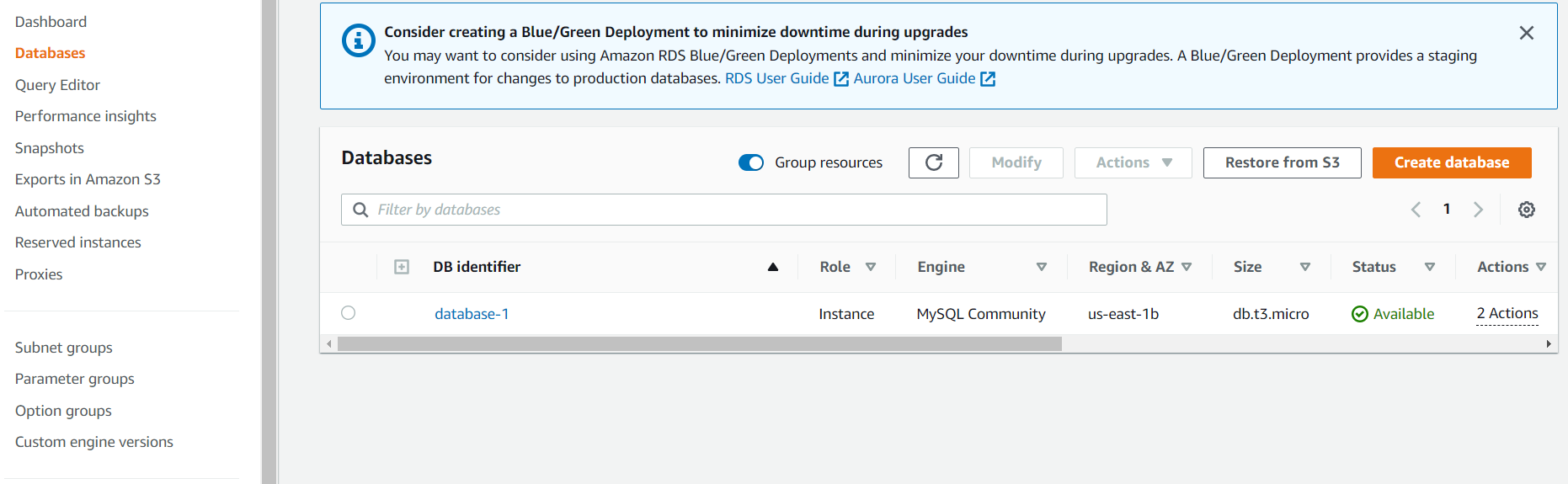
AWS RDS:

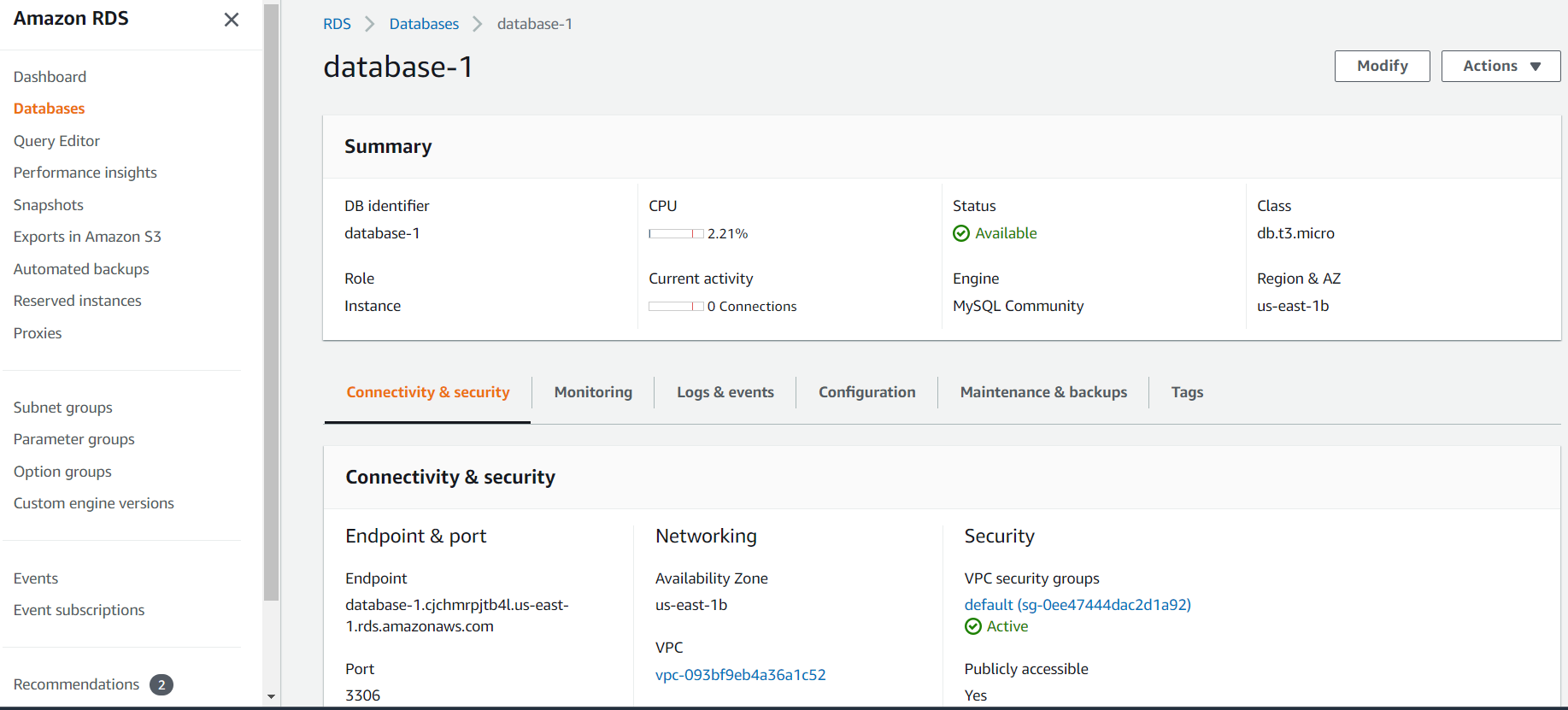
Amazon RDS is one of the web services provided by AWS that simplifies the setup, operation, and scaling of relational databases in the cloud. By automating many aspects of relational database management, Amazon RDS reduces administrative overhead. Additionally, Amazon RDS creates multiple instances to ensure high availability and enable failovers. The service supports several relational database engines, including PostgreSQL, MySQL, MariaDB, Oracle, SQL Server, and Amazon Aurora.

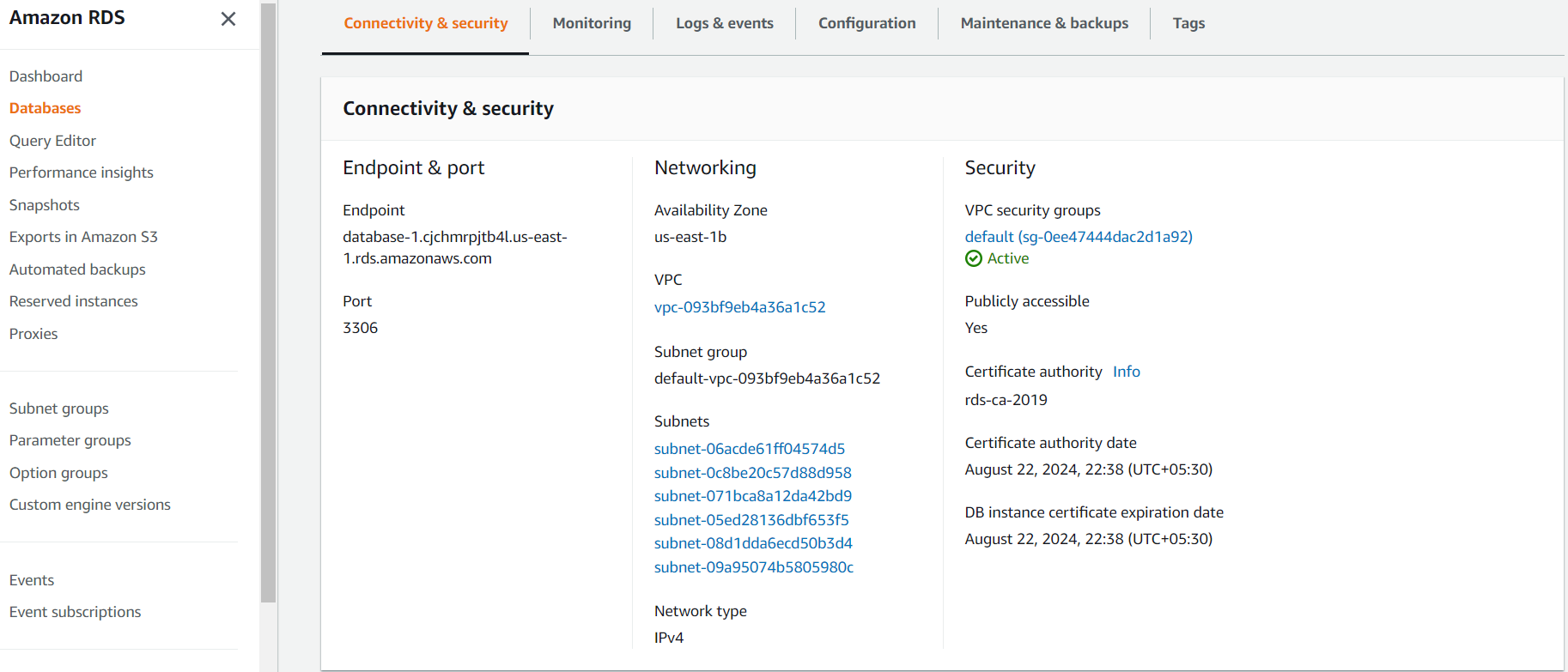
Amazon RDS stores data in a relational format, using tables, records, and fields. Tables can have relationships with one another, which is a fundamental aspect of relational databases. These types of databases are commonly used to store both transactional and analytical data. Relational databases offer high levels of stability and reliability, particularly transactional databases.

We used RDS in this project to store the data generated from the second webpage CONTACT. Rest API is invoked when the user tries to contact the ADMIN and then integrates with lambda to connect to the database which is MYSQL. The database connection/port details are mentioned in the lambda function using python.





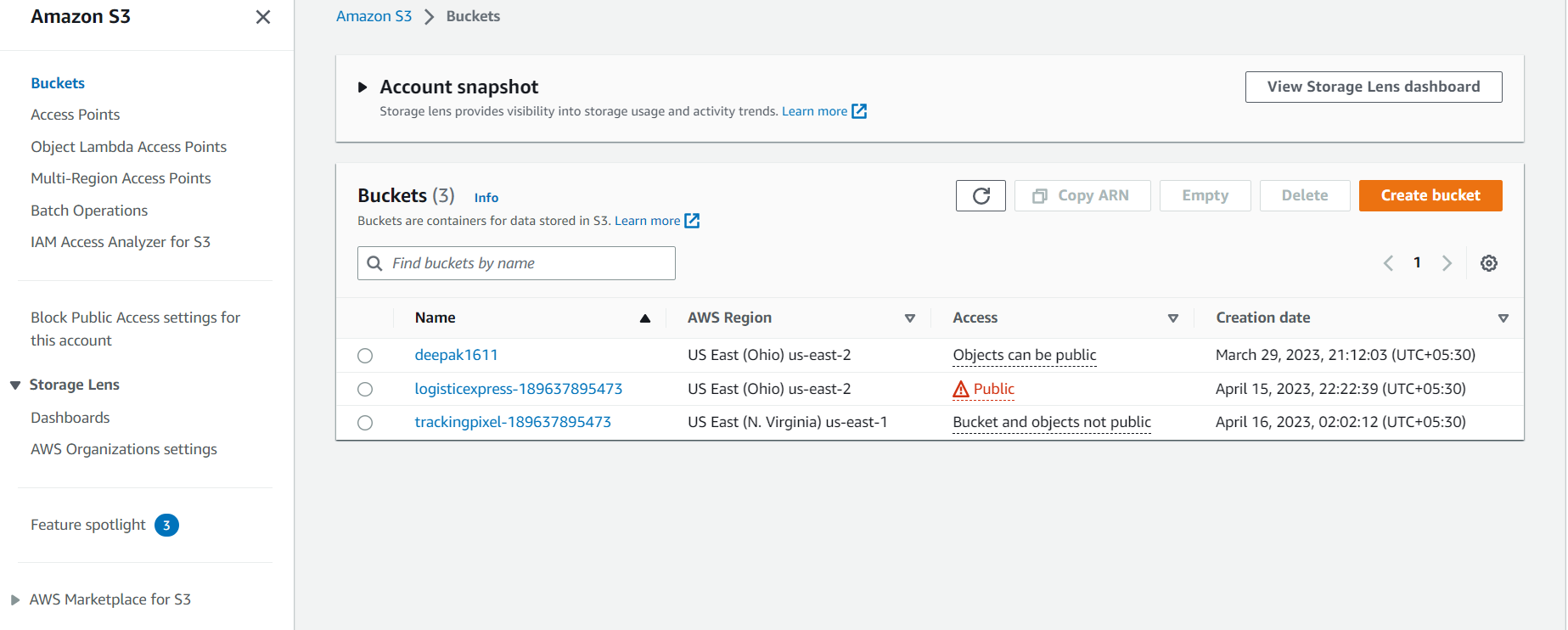


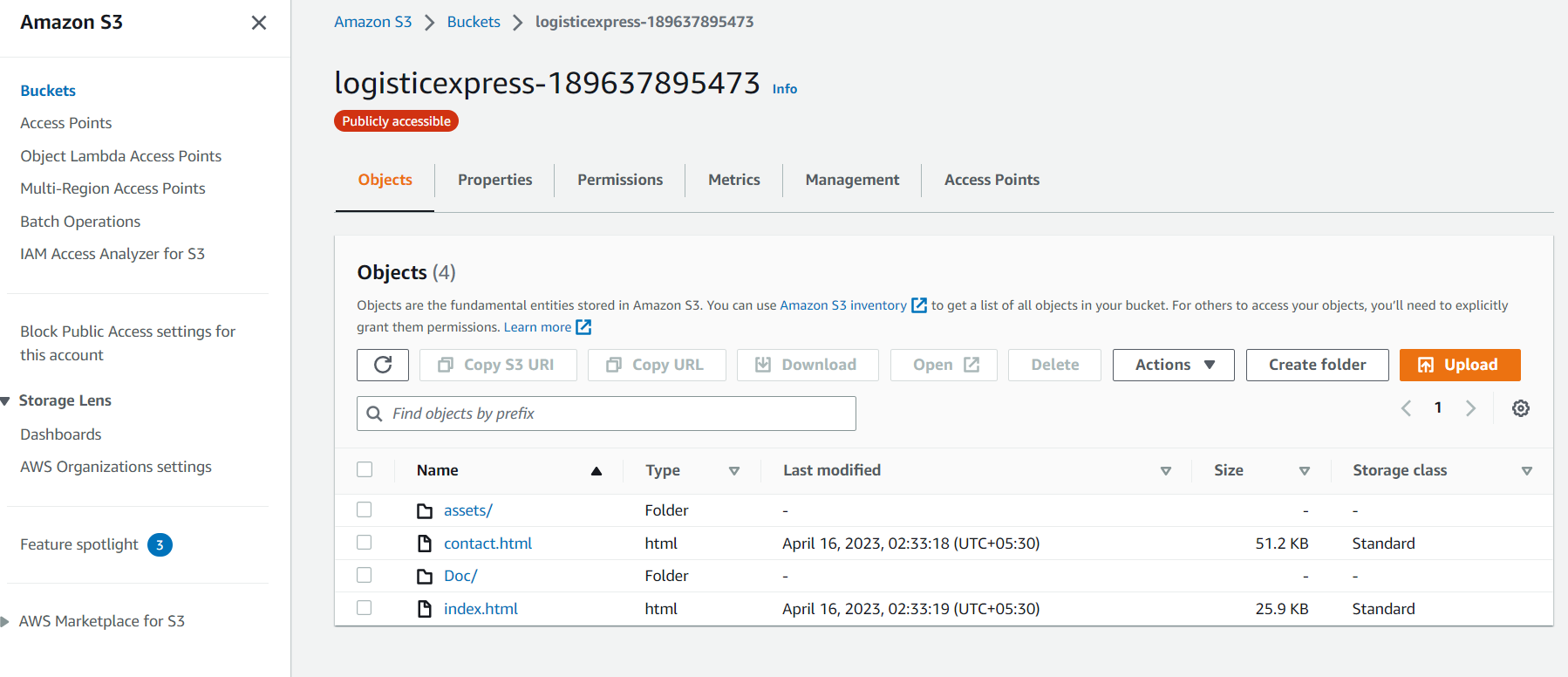


AWS S3:

Amazon S3 is a widely used object storage service offered by AWS, providing exceptional scalability, availability of data, enhanced security, and performance. Organizations of all types and sizes can leverage Amazon S3 to securely store and protect any volume of data for a diverse range of use cases, including data lakes, sites & mobile apps, backup & restore, archiving, enterprise apps, IoT devices, and big data analytics. With Amazon S3, you have access to a broad range of management features that allow you to enhance, systemize, and configure access to your data in a way that meets your unique business, organizational, and compliance requirements.

Simple storage service is an object storage service that provides highly secured and scalable storage which can be accessed from anywhere over the internet. In this project, the application is hosted on an S3 bucket. With the help of kinesis data firehose, which transforms and delivers the streaming data to the destination i.e., user data is pushed into this S3 bucket. In the S3 bucket, Where the data is classified into object files, We can simply download and verify the IP address, browser, and OS details of the user system and track the details of the visited page.





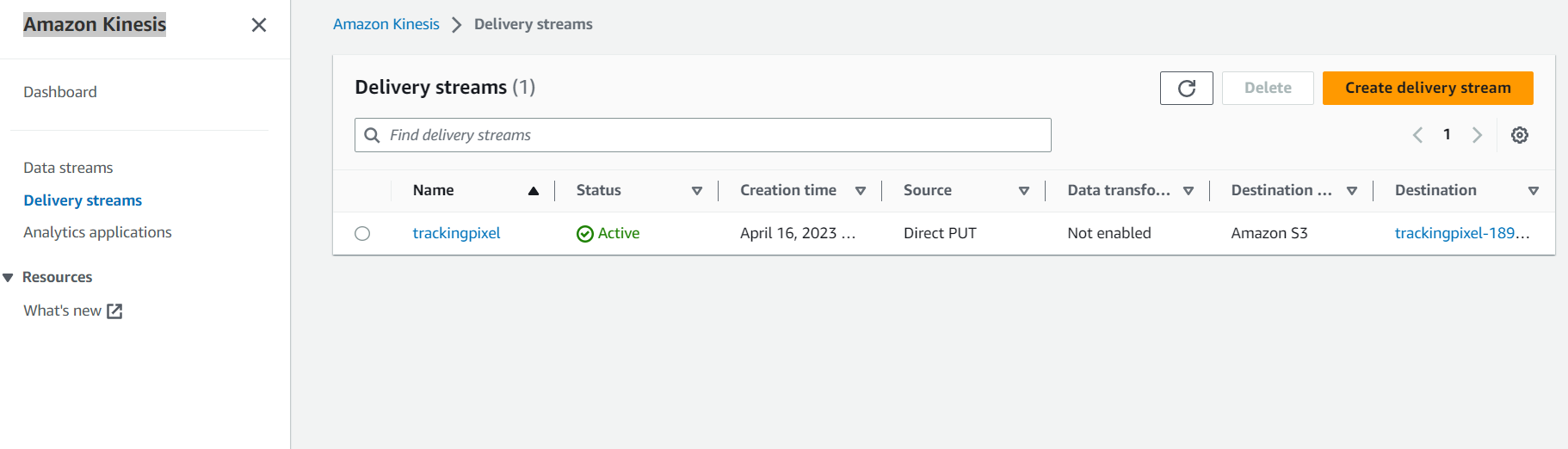
AWS Kinesis:

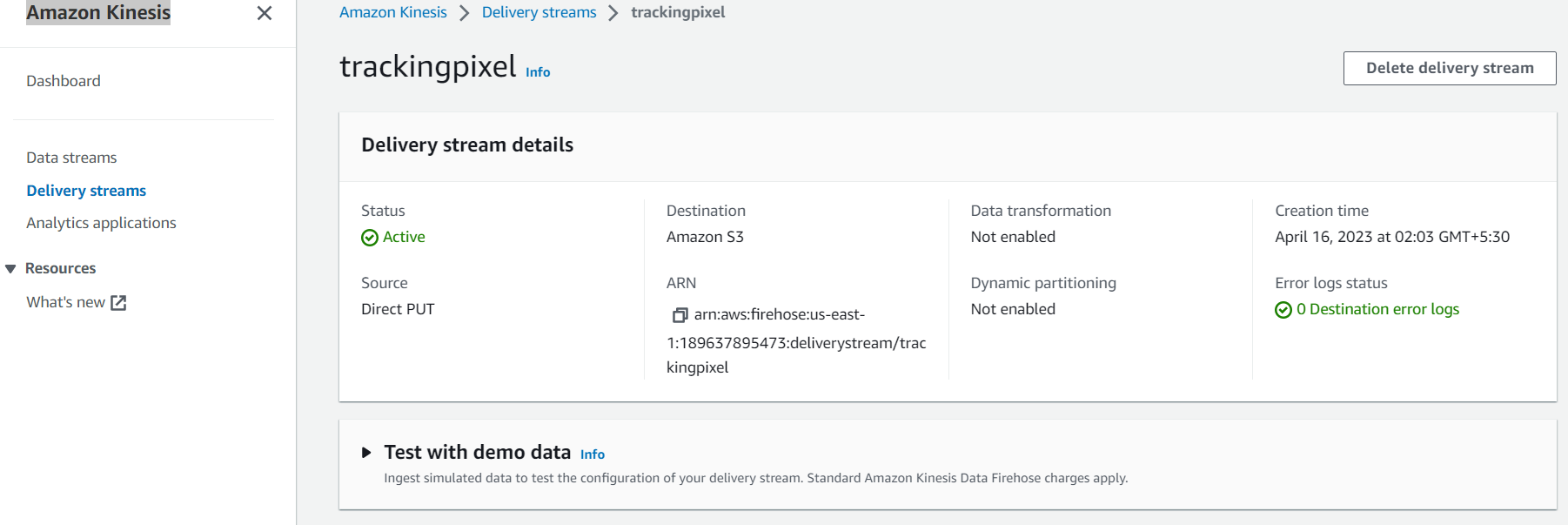
Amazon Kinesis is a highly scalable, cloud-based service that enables real-time processing of large volumes of streaming data per second. It is specifically designed for real-time applications and provides developers with the ability to ingest any amount of data from multiple sources while scaling up or down as needed.

Kinesis can record, save, and analyze data from distributed streams, such as activity logs i.e., event logs and online feeds. Once processed, Kinesis can distribute the data to multiple consumers simultaneously, making it a highly efficient tool for real-time data processing and analysis.

Amazon Kinesis is also a serverless service that captures large/small-scale data and then processes it and stores it for future usage. This data can then be utilized for analyzing and making business strategies.

Here when the user visits the webpage it triggers the HTTP API which then connects to the lambda. Lambda then sends the information to kinesis for data processing. Kinesis now pushes the data to the S3 bucket. Lambda is configured in the Delivery streams and the destination for data storage is specified as an S3 bucket.





User Interface: