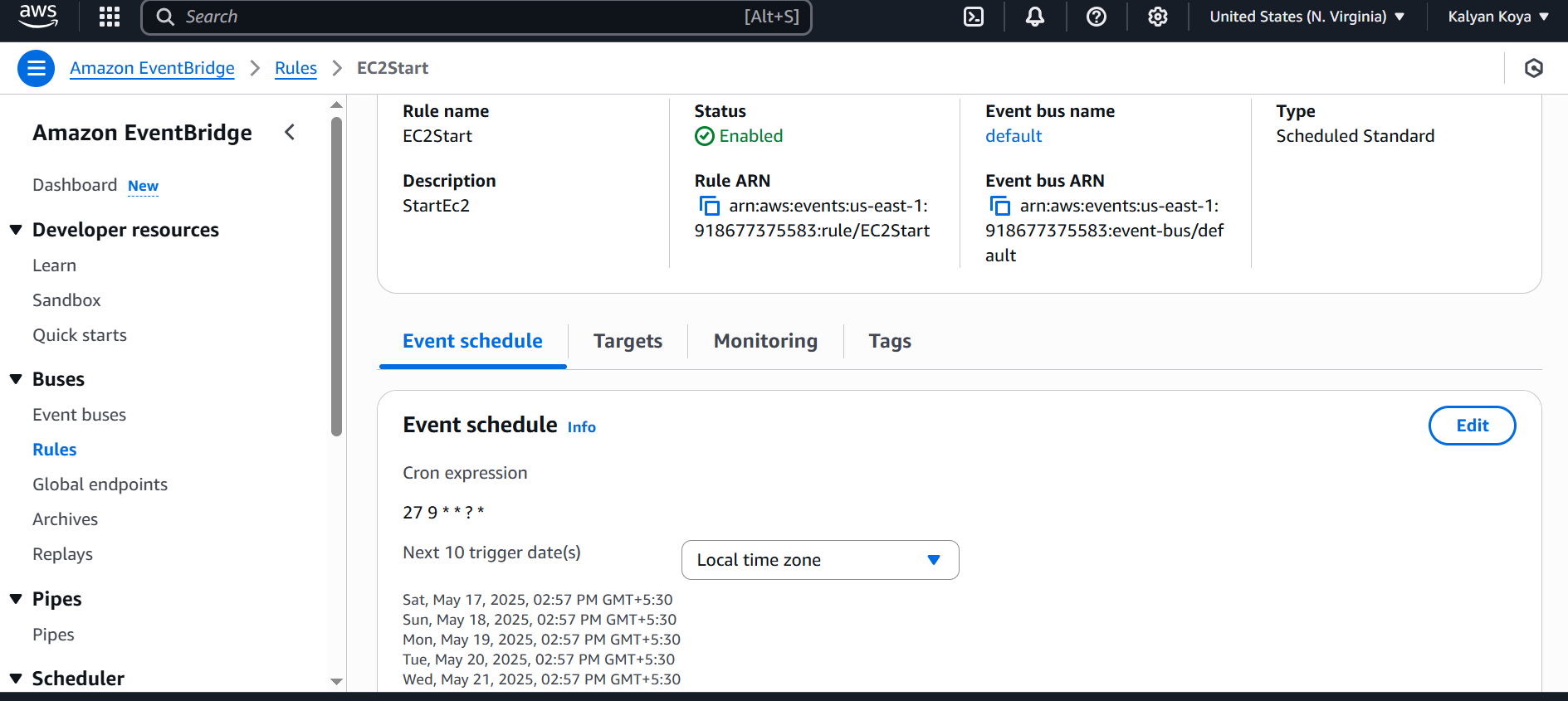
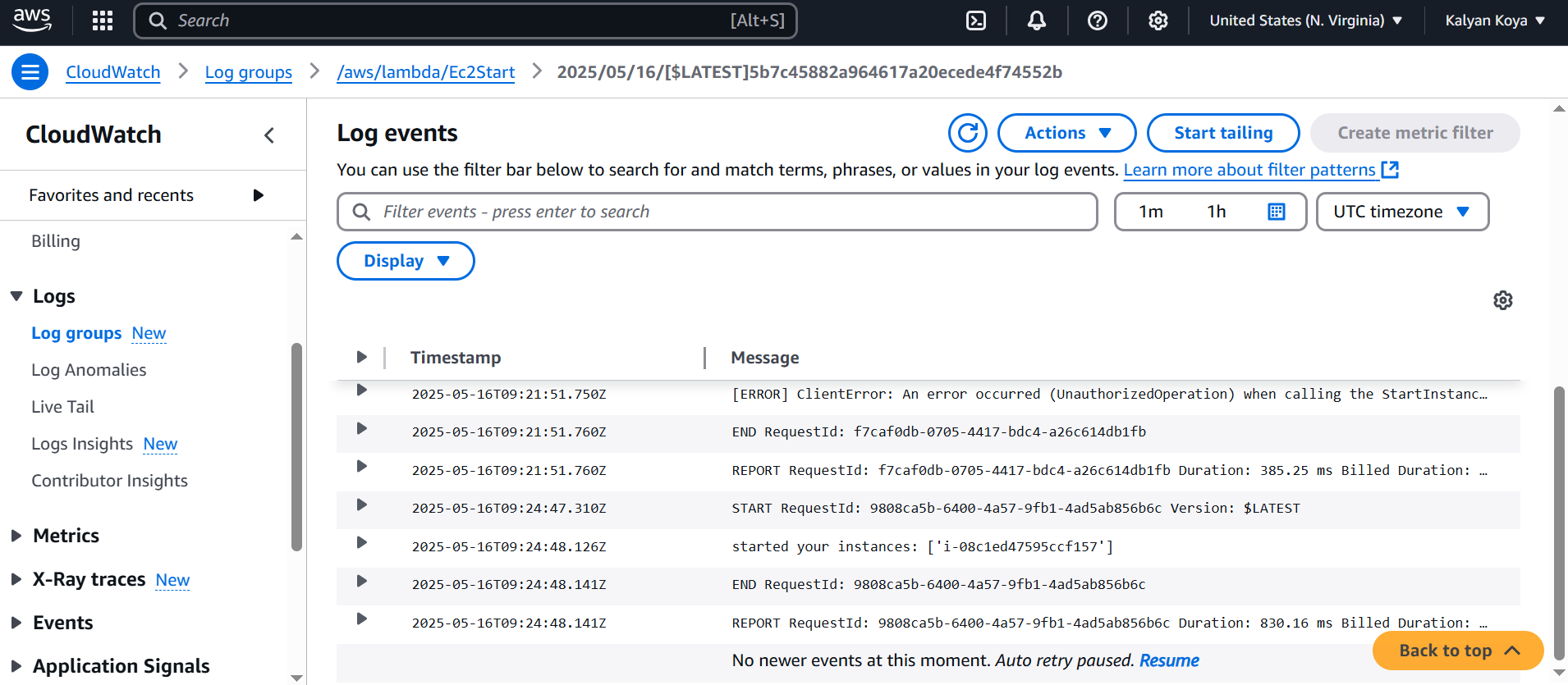
1. **Event bridge: -- Invoke Lambda function to stop and start through Cron Jobs.**

**Rules to start and attached lambda function**



**Cloudwatch logs for starting**



**2) Amazon Simple Queue Service (SQS)**

**What is SQS?** Amazon SQS is a fully managed message queuing service designed to decouple components of a distributed system. It allows different parts of an application to communicate by sending messages via a queue. This decoupling makes applications more resilient, scalable, and easier to maintain. Because SQS supports both standard (with best-effort ordering and at-least-once delivery) and FIFO (first-in, first-out, exactly-once processing) queues, it can be tailored to various workload requirements.

**Real-Time Example / Case Study:** Consider an e-commerce platform where users can upload images for their product listings. When a user uploads an image, the process of image processing (resizing, filtering, watermarking) is offloaded from the primary web server. Instead, a message – detailing what needs to be done – is placed into an SQS queue. Separate worker processes (or AWS Lambda functions) then poll this queue to fetch, process, and store the image. This asynchronous workflow enables the system to handle spikes in user uploads without overwhelming the processing logic, thus ensuring a smooth user experience and improved system reliability. Another popular case is a live voting system during televised events. When millions of votes may be cast in seconds, SQS absorbs the voting requests in a scalable, distributed manner, decoupling data ingestion from the vote tallying process and thereby managing significant load spikes efficiently.

**3) Amazon Simple Notification Service (SNS)**

**What is SNS?** Amazon SNS is a fully managed publish-subscribe (pub-sub) messaging service. It’s designed to broadcast messages to multiple subscribers across different protocols, including HTTP/S, email, SMS, and push notifications. Unlike SQS, which deals with point-to-point messaging, SNS’s strength lies in its ability to fan out messages to a multitude of endpoints simultaneously, enabling real-time notifications and agile, event-driven communication.

**Real-Time Example / Case Study:** Imagine a news agency mobile application that must instantly notify its users of breaking news. Using SNS, the agency can publish a message to an SNS topic as soon as a news alert is issued. All mobile devices, emails, or SMS endpoints subscribed to that topic receive the notification within seconds—ensuring that users are kept up-to-date in real time. Similarly, consider an online retail system that sends out notifications via multiple channels (SMS, email, and mobile push) when there’s an update on order status. This multi-channel, real-time broadcasting capability is at the heart of SNS, making it an ideal solution for applications where immediate dissemination is critical 4.

**4) AWS Step Functions and State Machines**

**What are Step Functions and a State Machine?** AWS Step Functions is a service that enables the orchestration of complex workflows by coordinating different AWS services into serverless workflows. At the core of Step Functions is the concept of a **state machine**, which is a model used to define a sequence of steps (states) where each state represents a distinct task (such as invoking a Lambda function, making a choice based on input, or handling errors). These state machines not only provide a visual map of the workflow but also integrate built-in error handling and retries, making it easier to manage distributed executions.

**Real-Time Example / Case Study:** A classic example is an e-commerce order processing system. When a customer places an order, the system must validate the order, charge the customer’s card, update inventory, and finally send an order confirmation email. Each of these steps can be represented as states in a state machine:

* **Order Validation:** Ensure that all product details and customer information are correct.
* **Payment Processing:** Charge the credit card using a payment gateway.
* **Inventory Update:** Adjust the stock levels based on the order.
* **Notification:** Send an order confirmation to the customer.

Using Step Functions, this entire process is orchestrated as a single workflow. If one particular state fails (for example, if the payment processing encounters an error), the state machine can automatically retry the operation or trigger compensating functions to rollback earlier steps. Another case study involves a serverless video encoding pipeline where video uploads trigger a workflow that transcodes the video, generates thumbnails, and stores metadata—all coordinated through a state machine. This pattern ensures that each stage operates reliably and that the entire process can be monitored, adjusted, or retried as needed.

These examples illustrate the principles and benefits of decoupling, scaling, and orchestrating distributed systems using AWS services. Whether you’re managing asynchronous tasks with SQS, sending real-time notifications with SNS, or coordinating complex workflows with Step Functions, each service offers unique advantages that help build robust, scalable architectures.