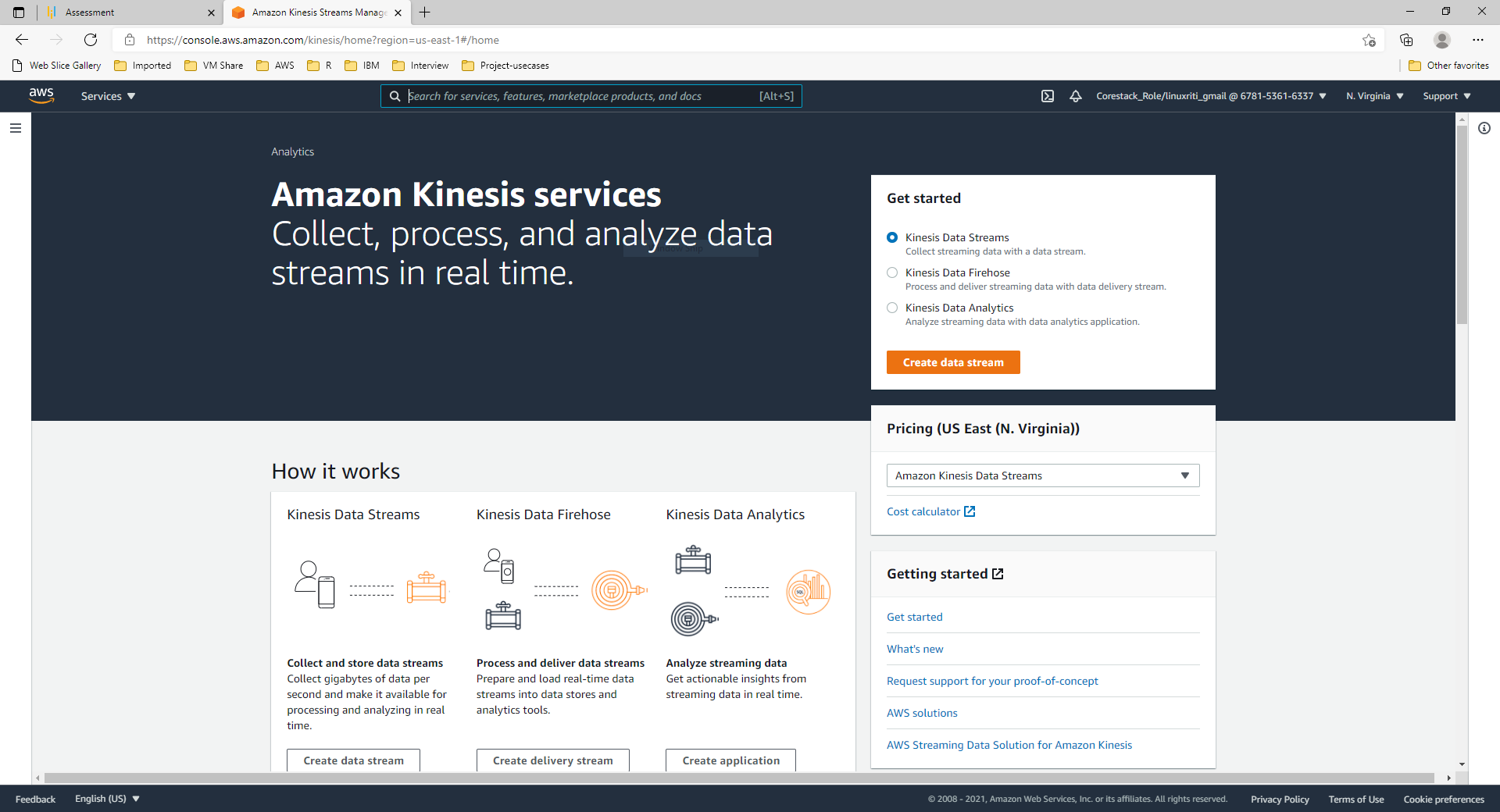
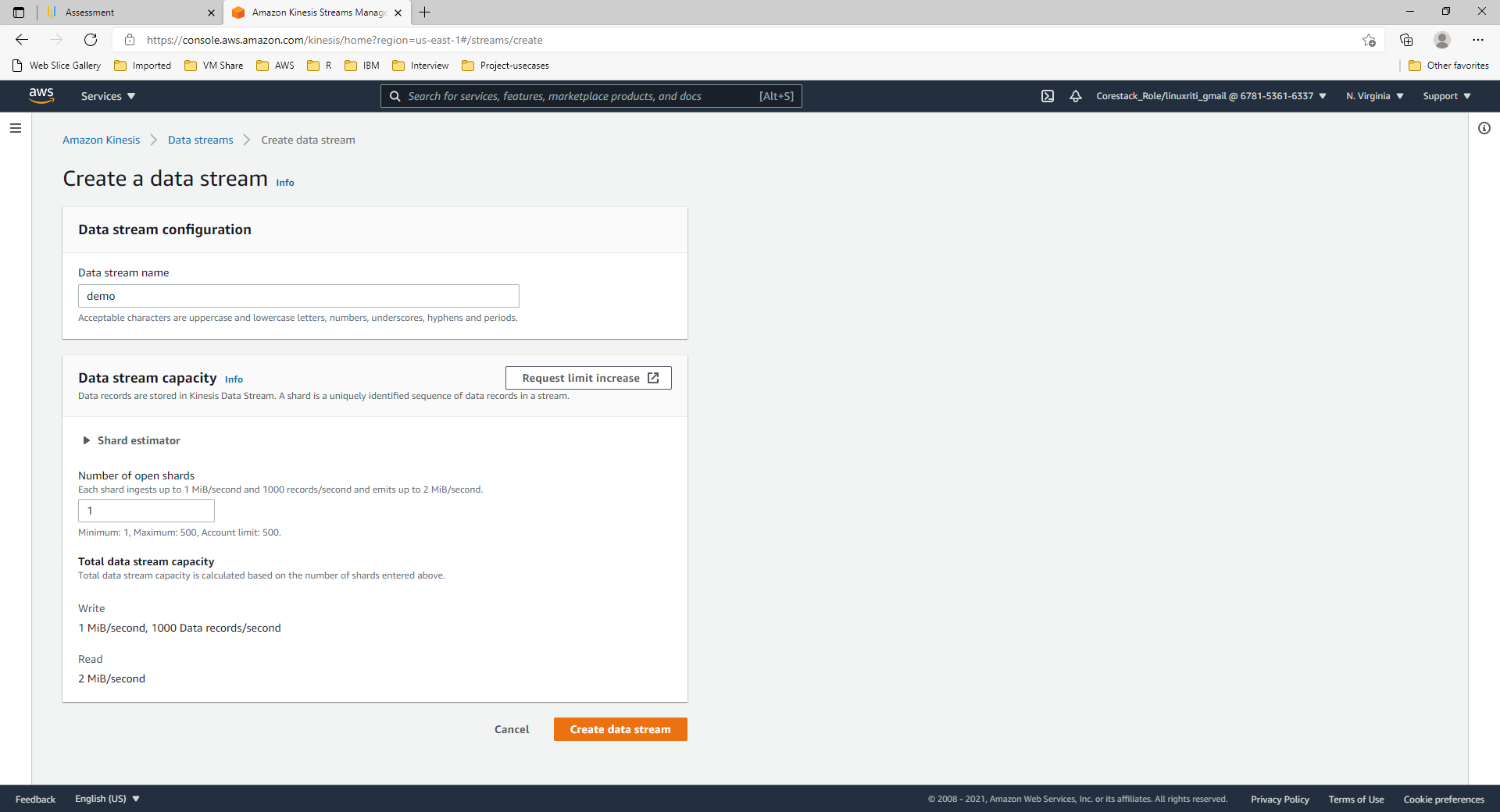
**Writeup:**

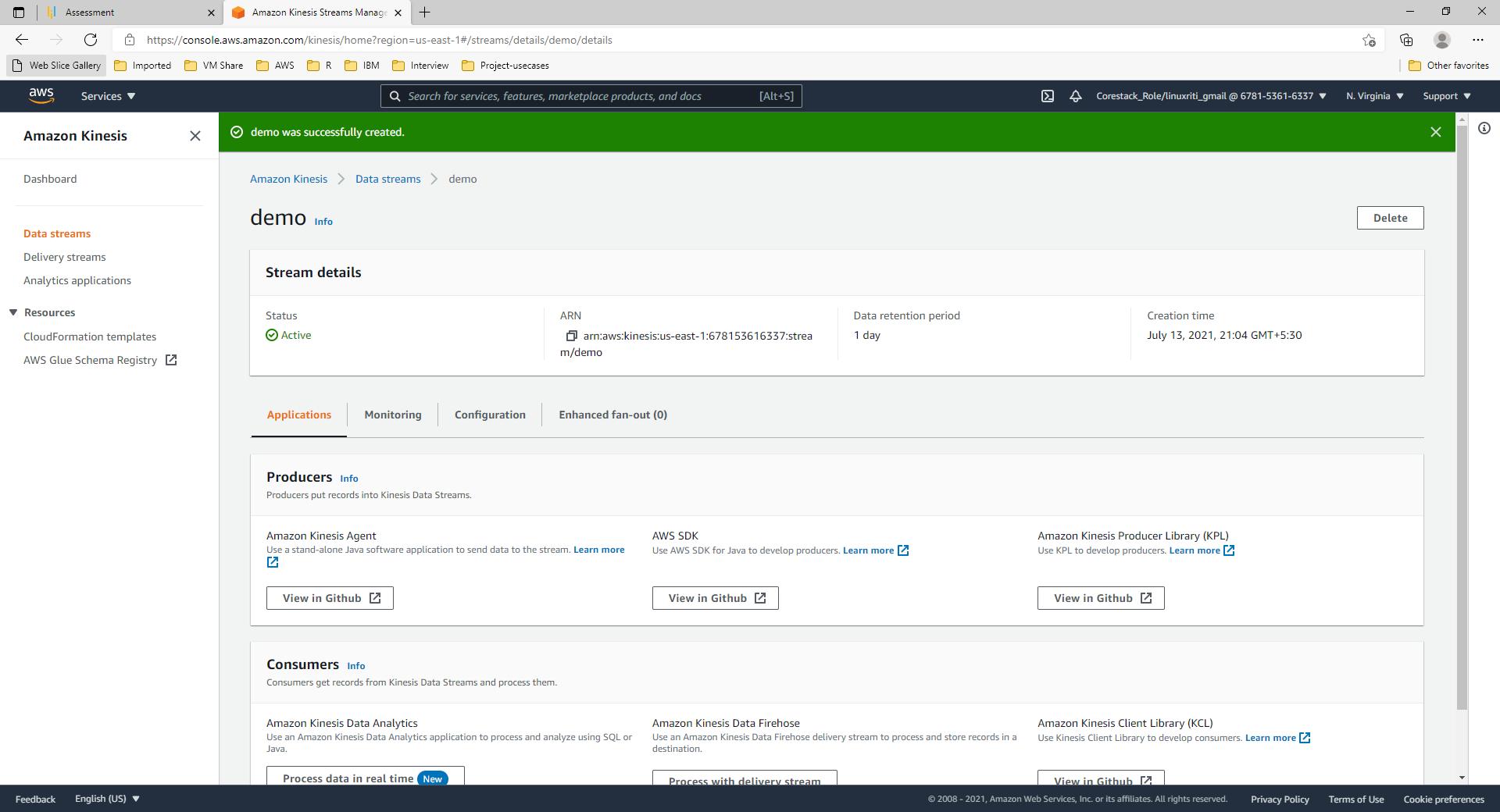
Step 1: Create Kinesis delivery stream

* + 1. I went to AWS console and open Kinesis. Then I created a data stream with name **‘Demo’**.

**Screenshot:**





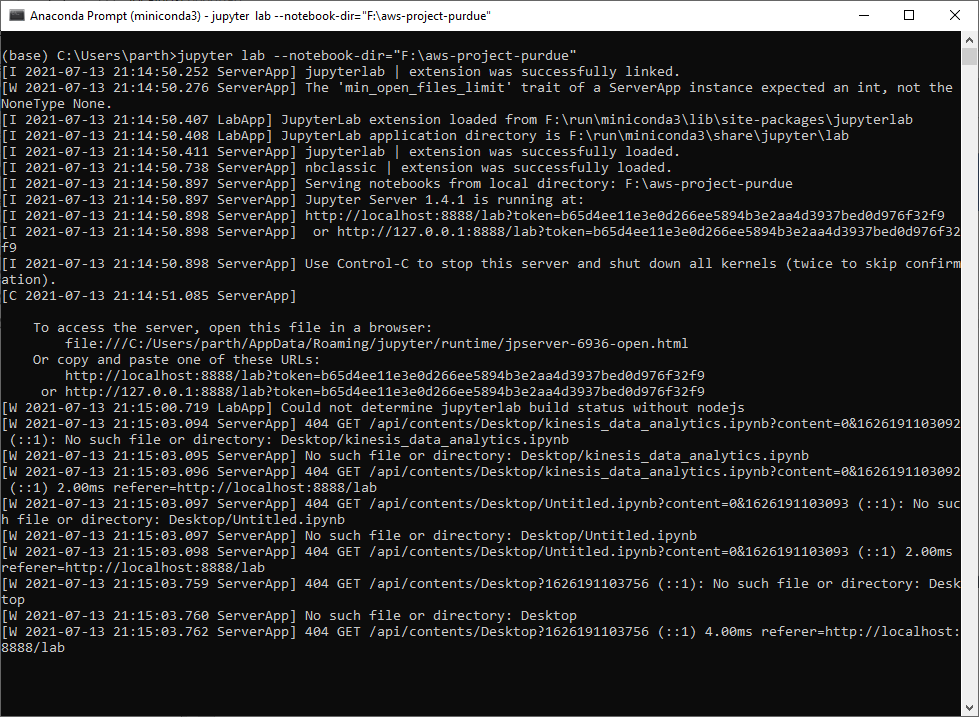


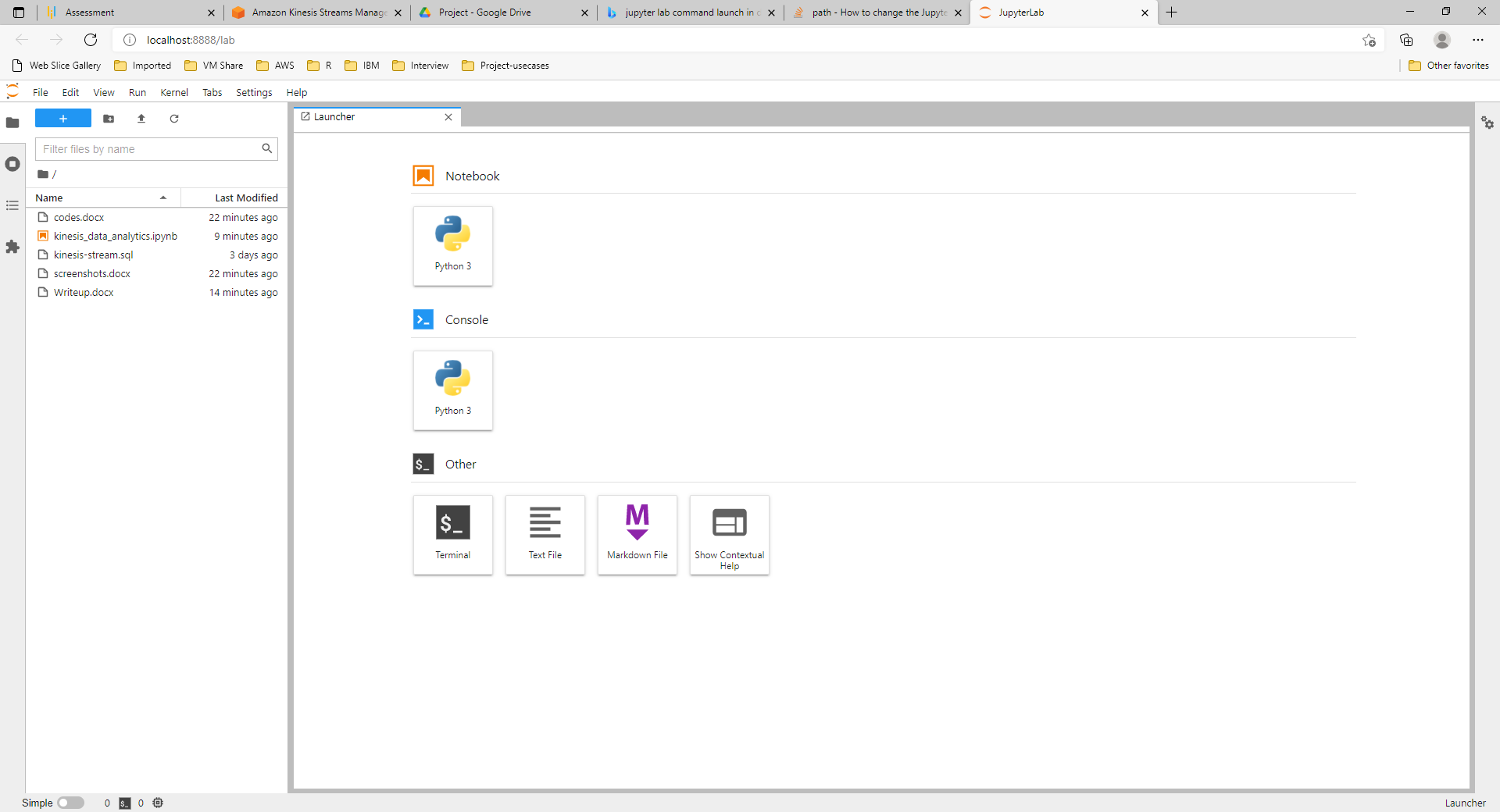
1. Install & launch miniconda3/ jupyter lab

a) installed miniconda3 and ran the ‘miniconda3 command prompt’ from start

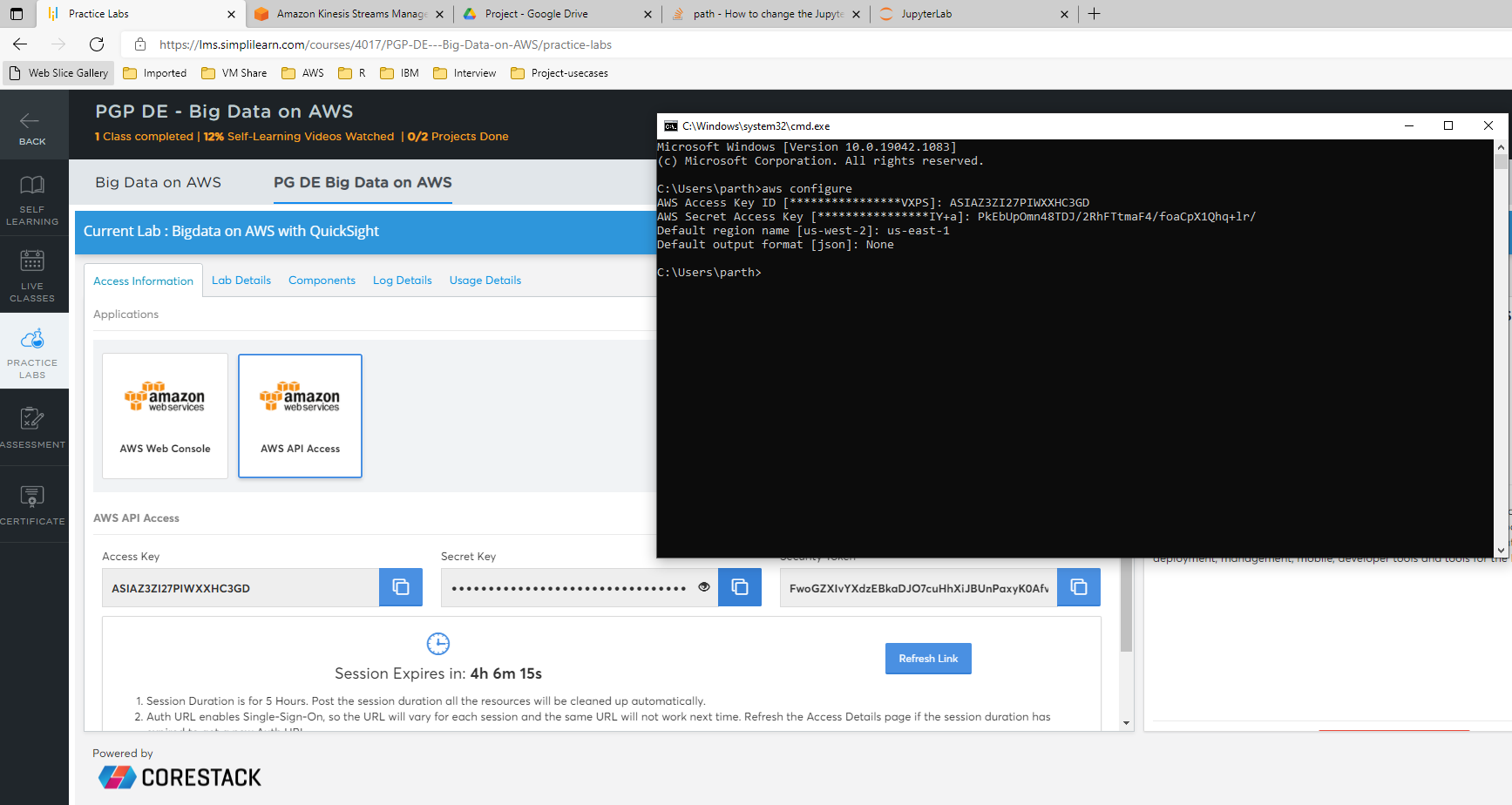
1. I gave the command jupyter lab --notebook-dir=‘F:\aws-project-purdue’ to launch jupyter lab.

Screenshot:

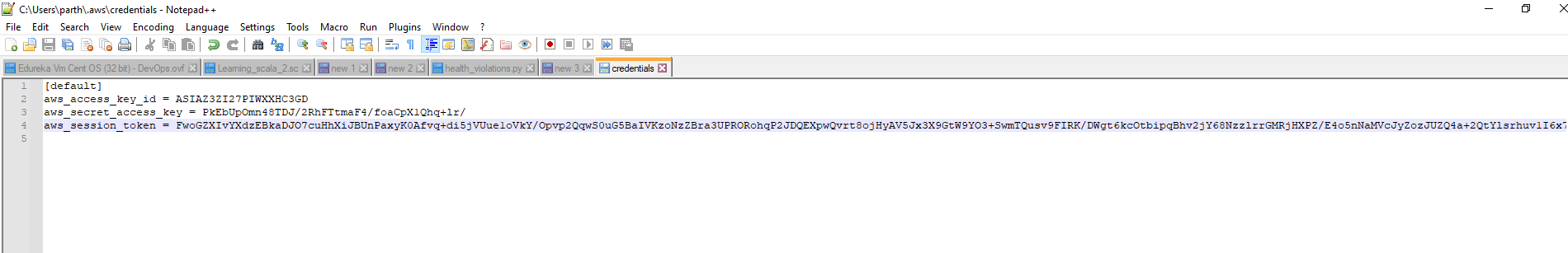




1. Install and configure AWSCLIV2
   1. aws config

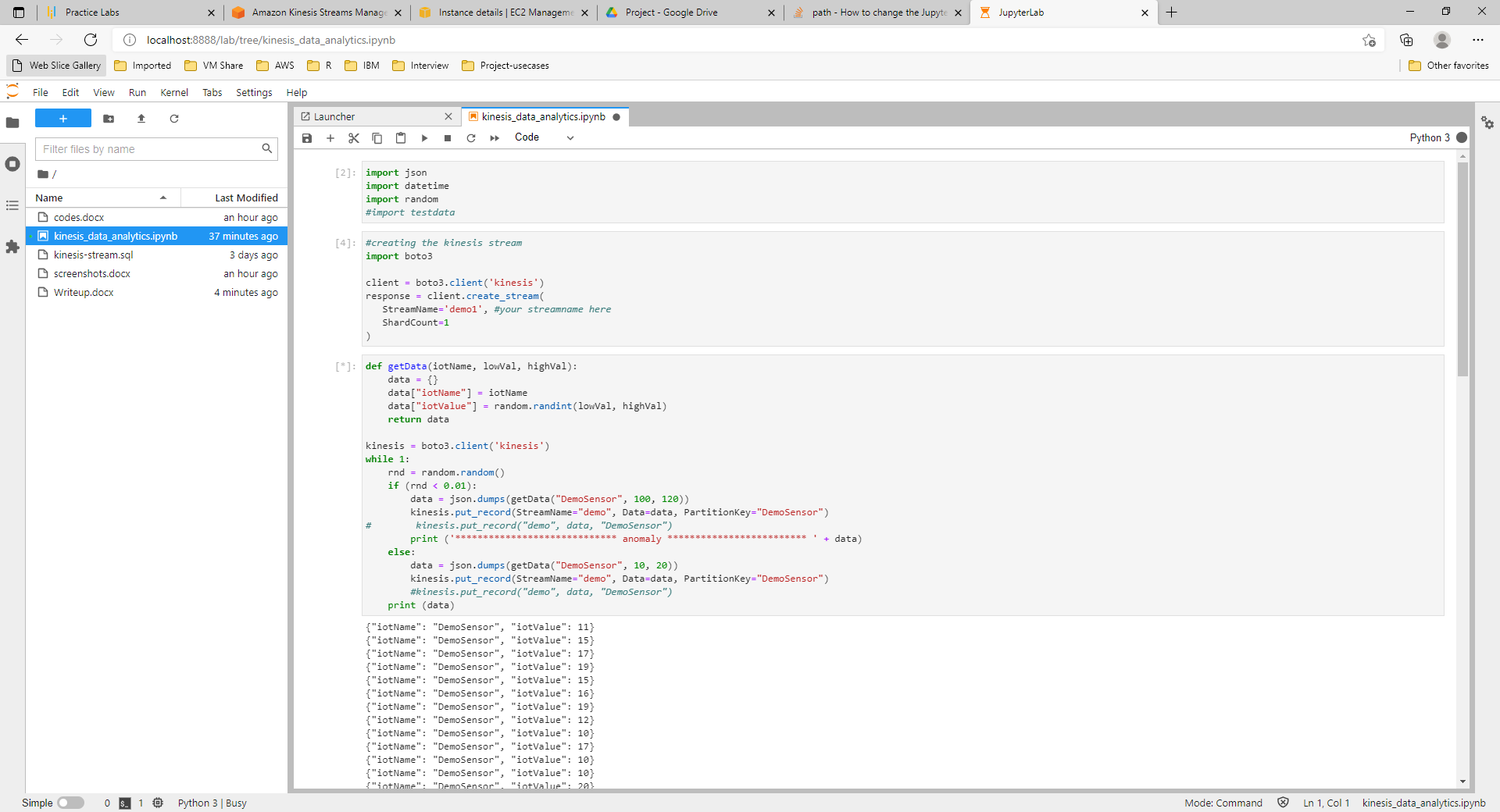


1. open file C:\Users\parth\.aws\credentials
   * 1. insert key aws\_session\_token = <paste the value from above screenshot>



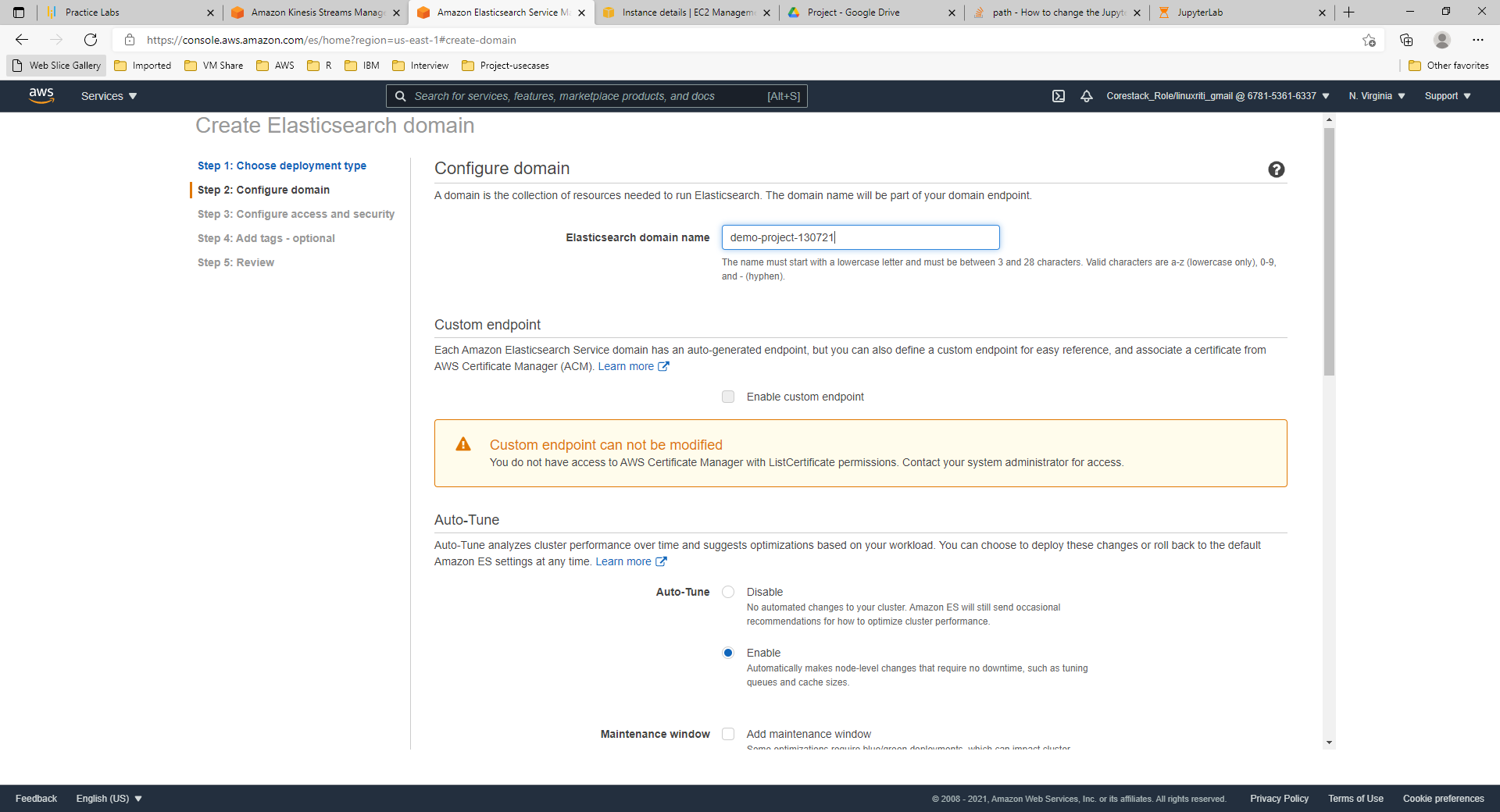
**Step 2: Simulate streaming application to detect anomalies**

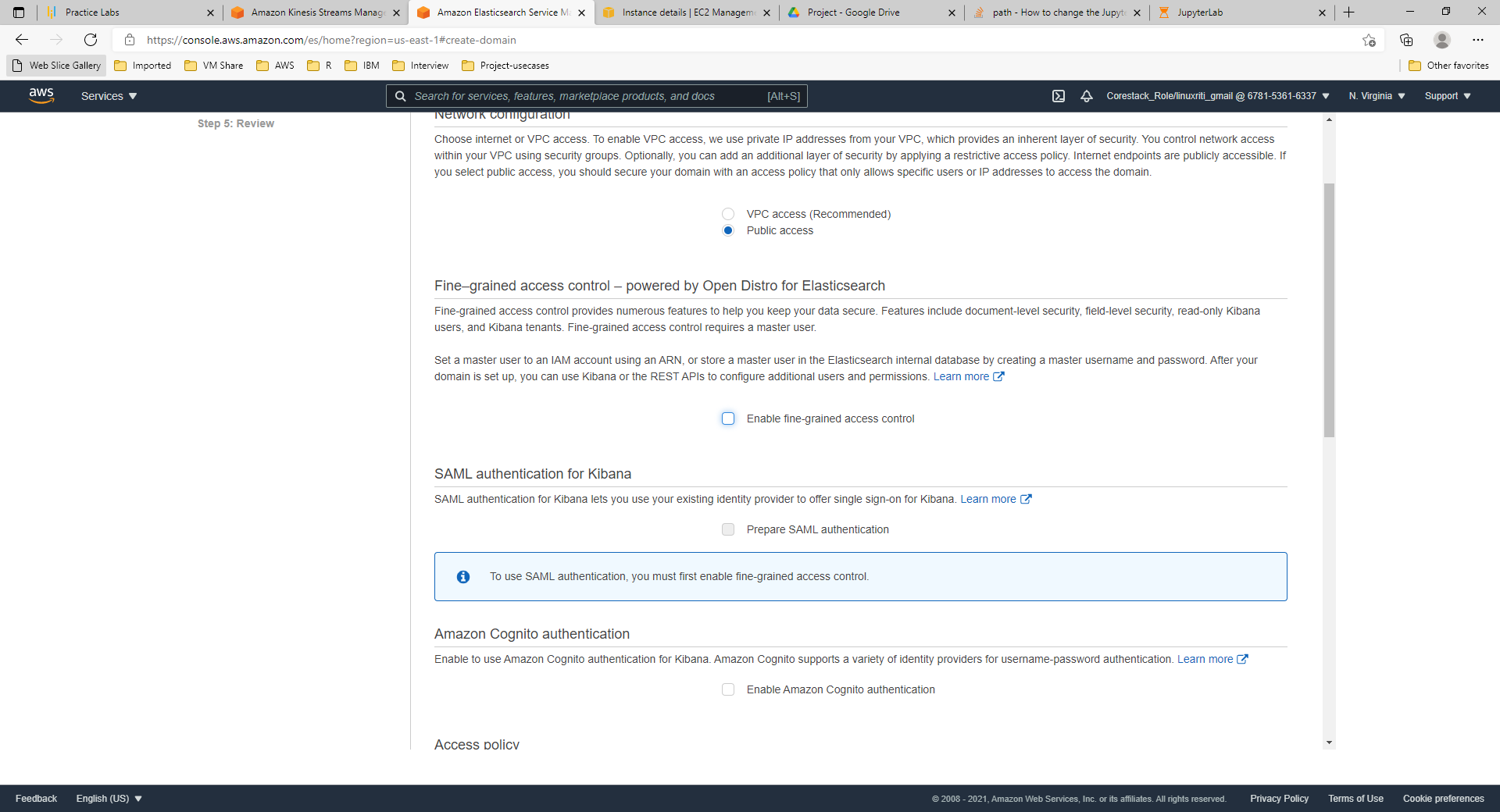
1. Run ‘kinesis\_data\_analytics.ipynb’ file in jupyter lab

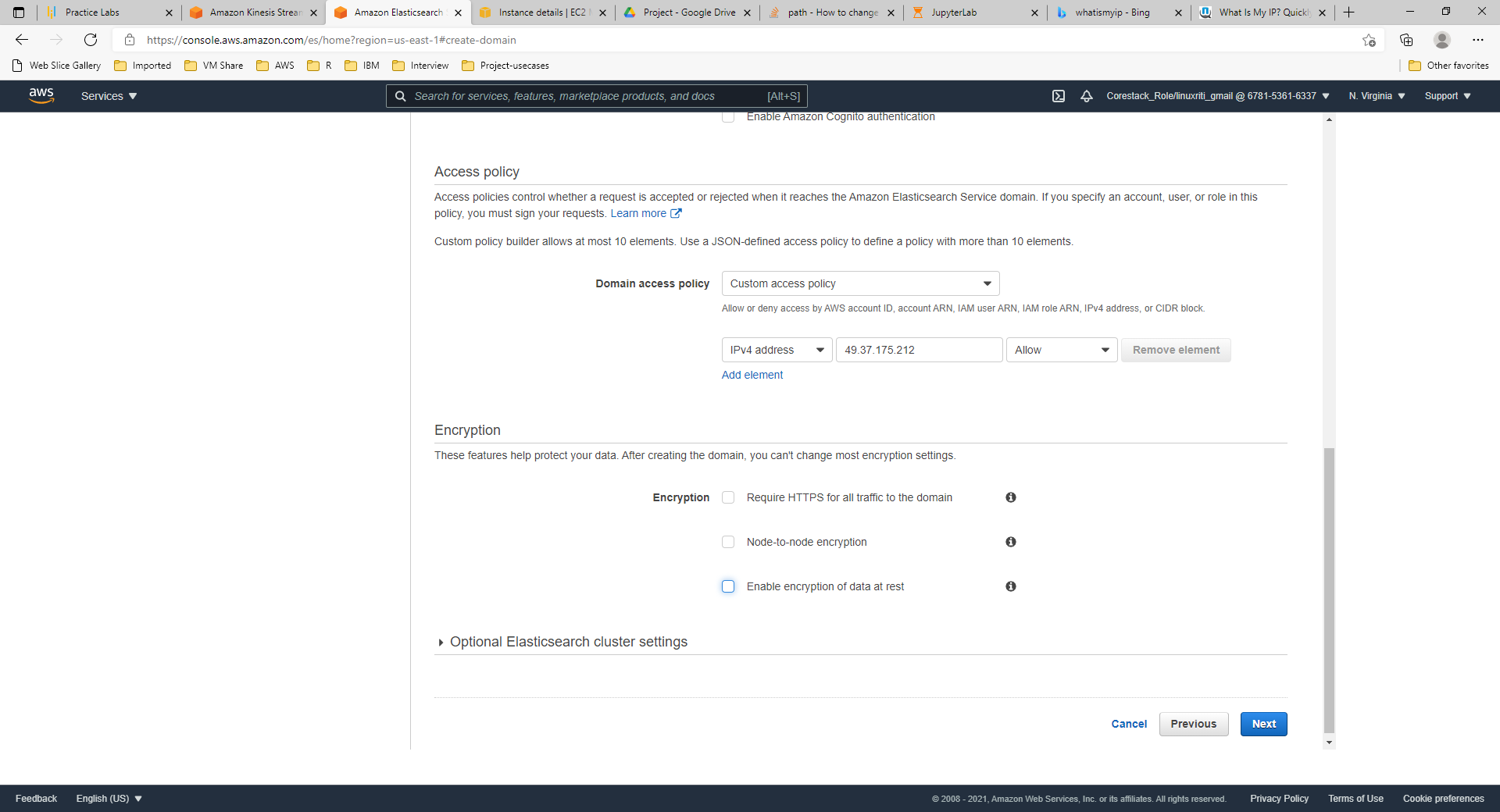


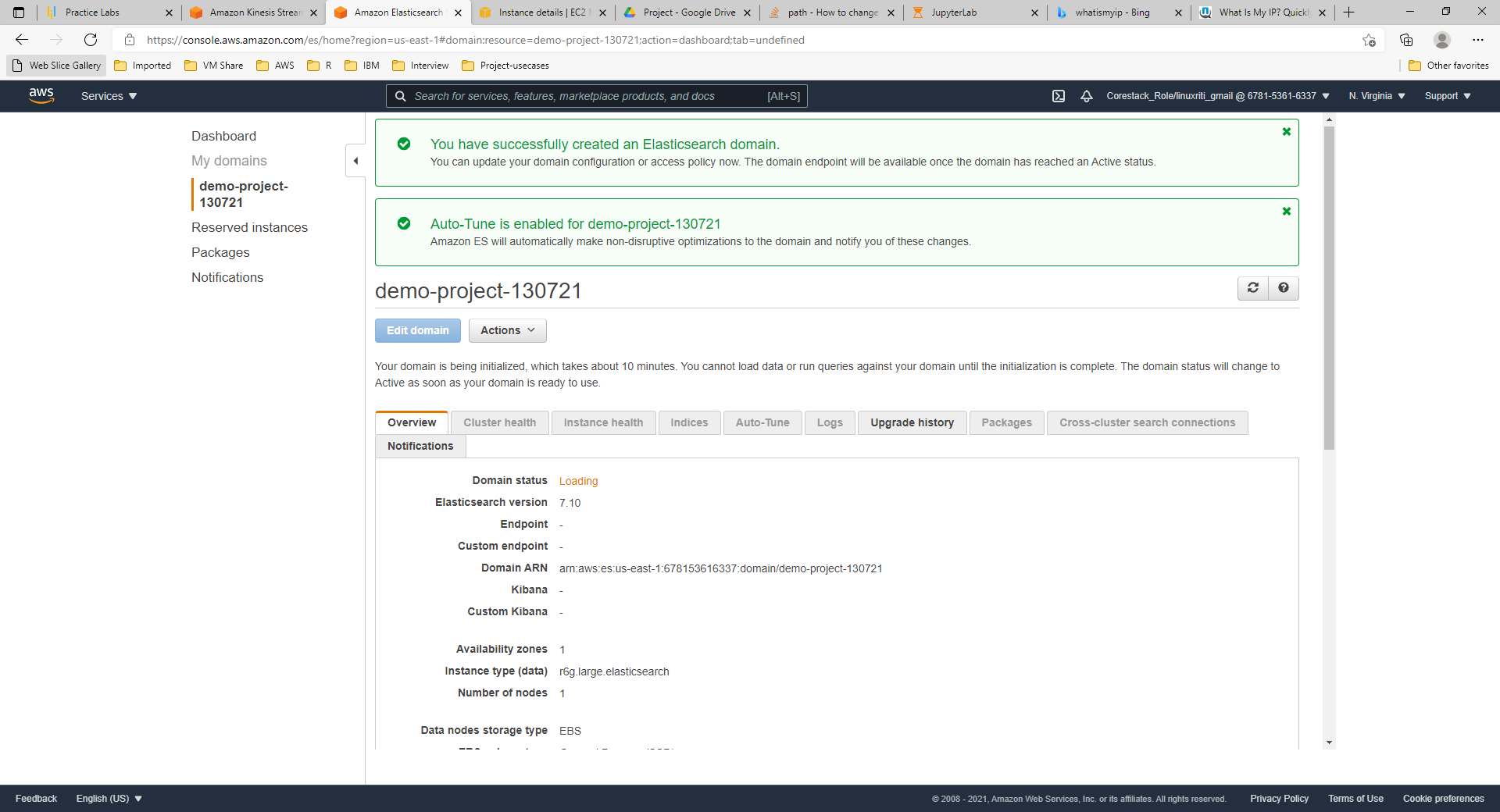
**Step 3: Open Elasticsearch service and create a new domain**



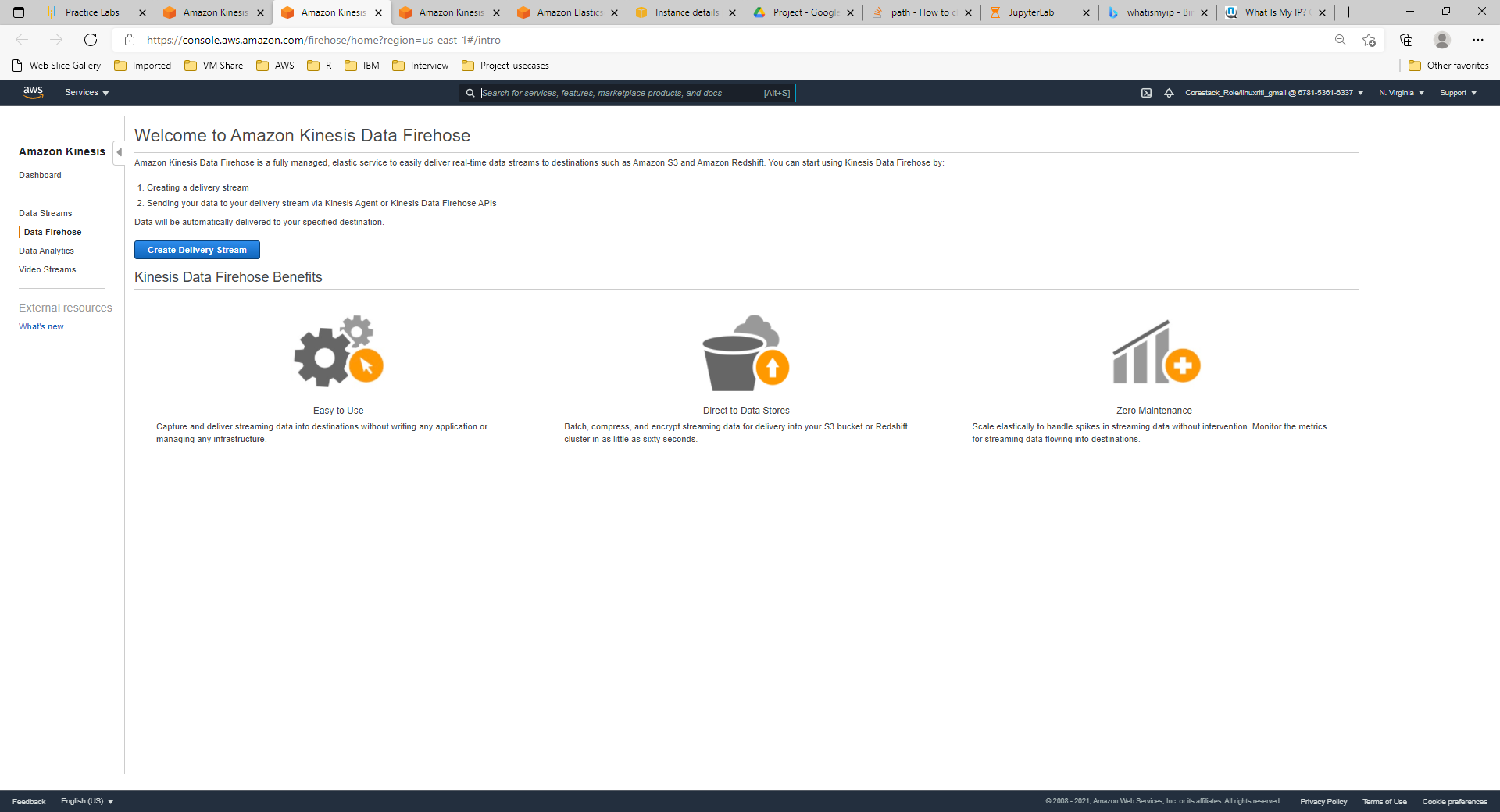


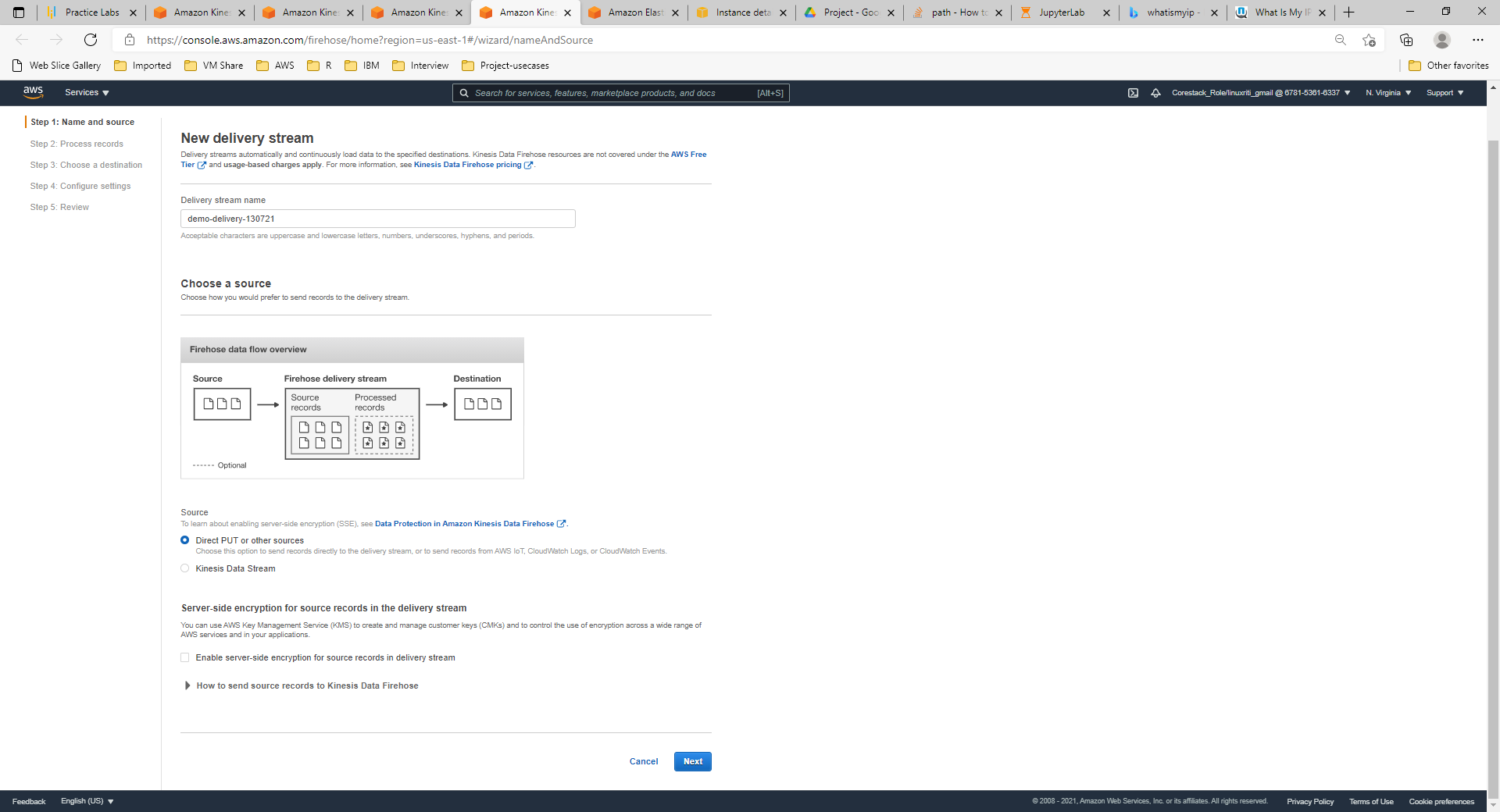


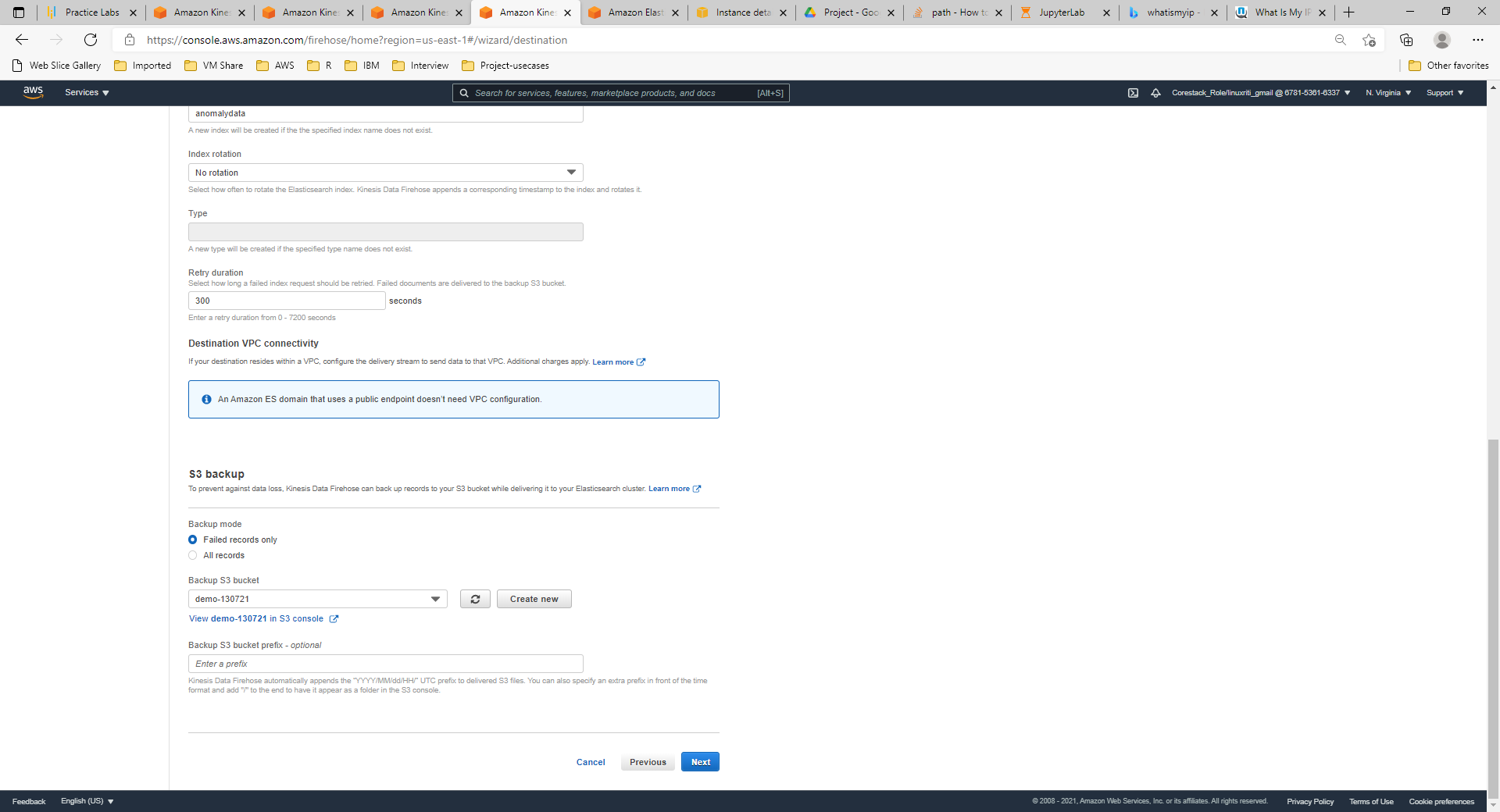


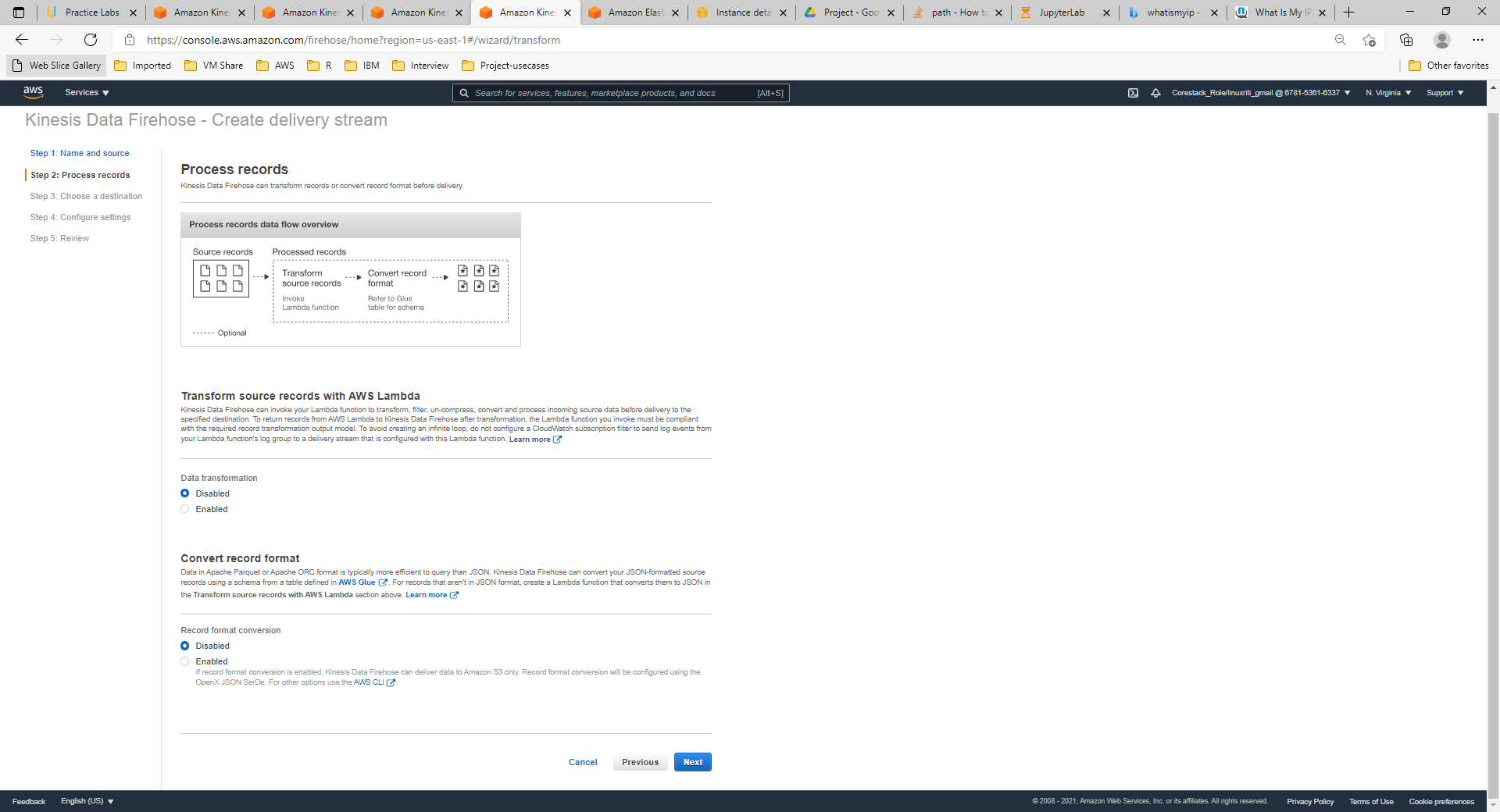


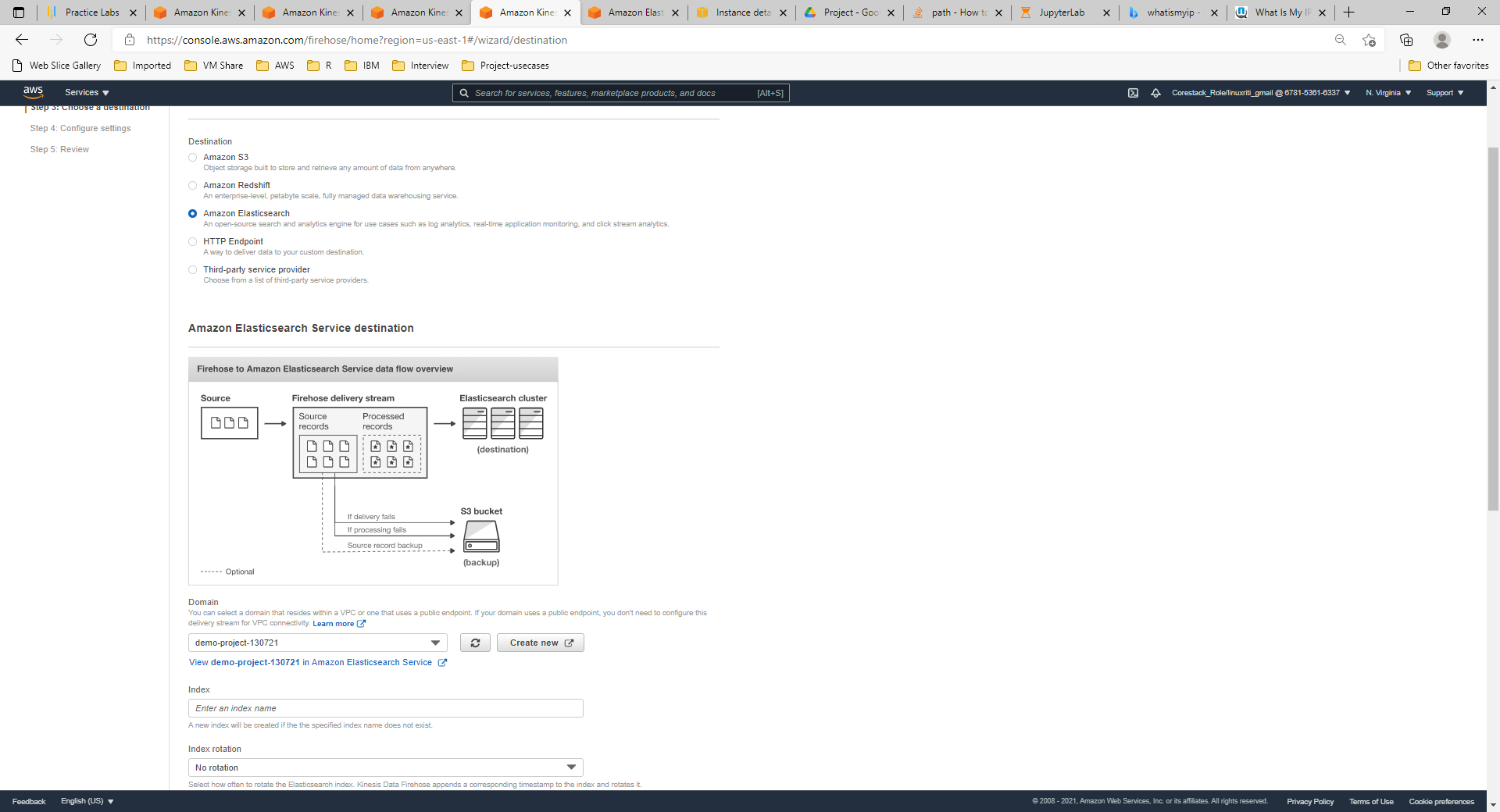
**Step 4: Configure Kinesis Firehose to export the results to Amazon ES**

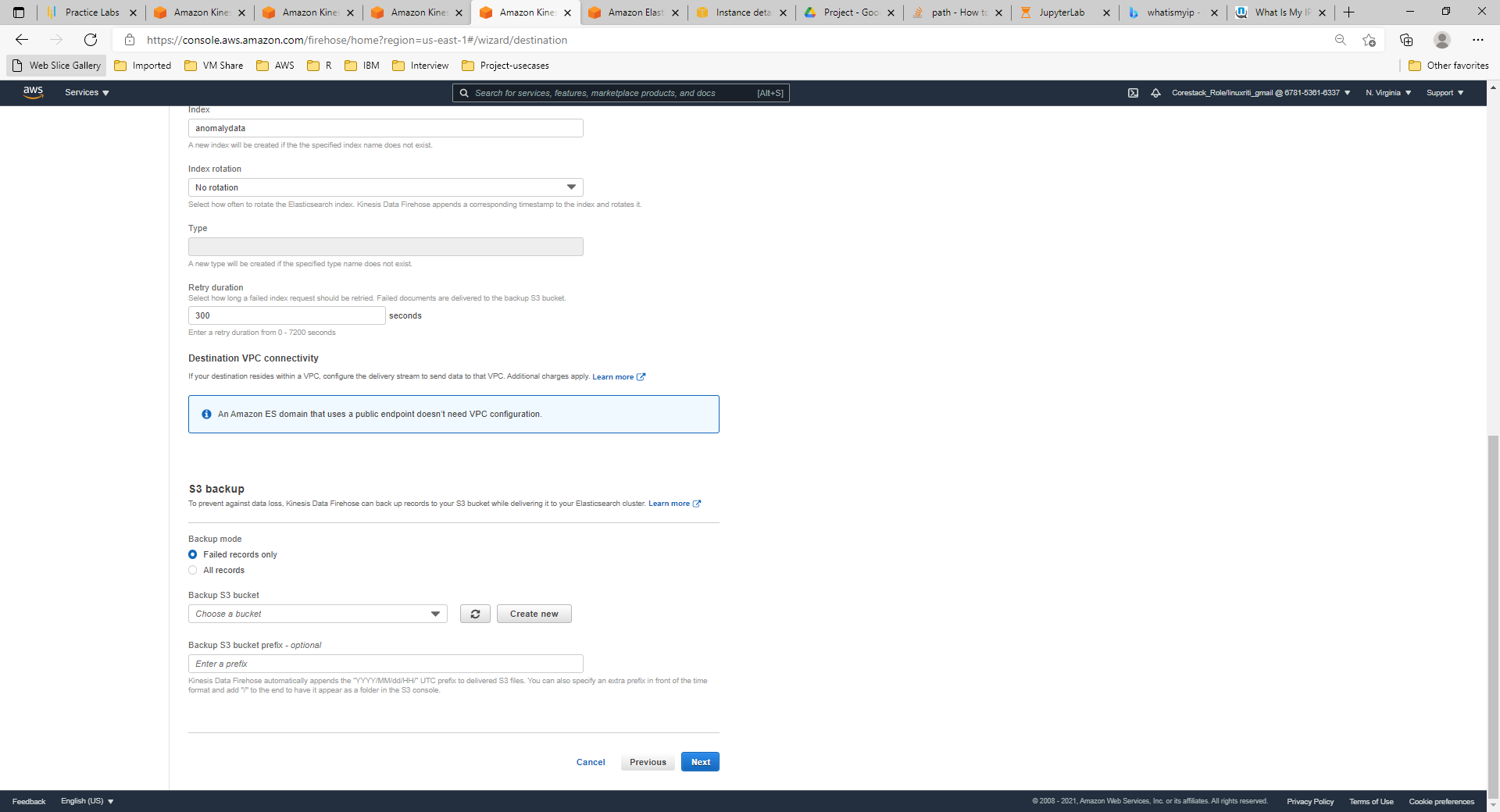




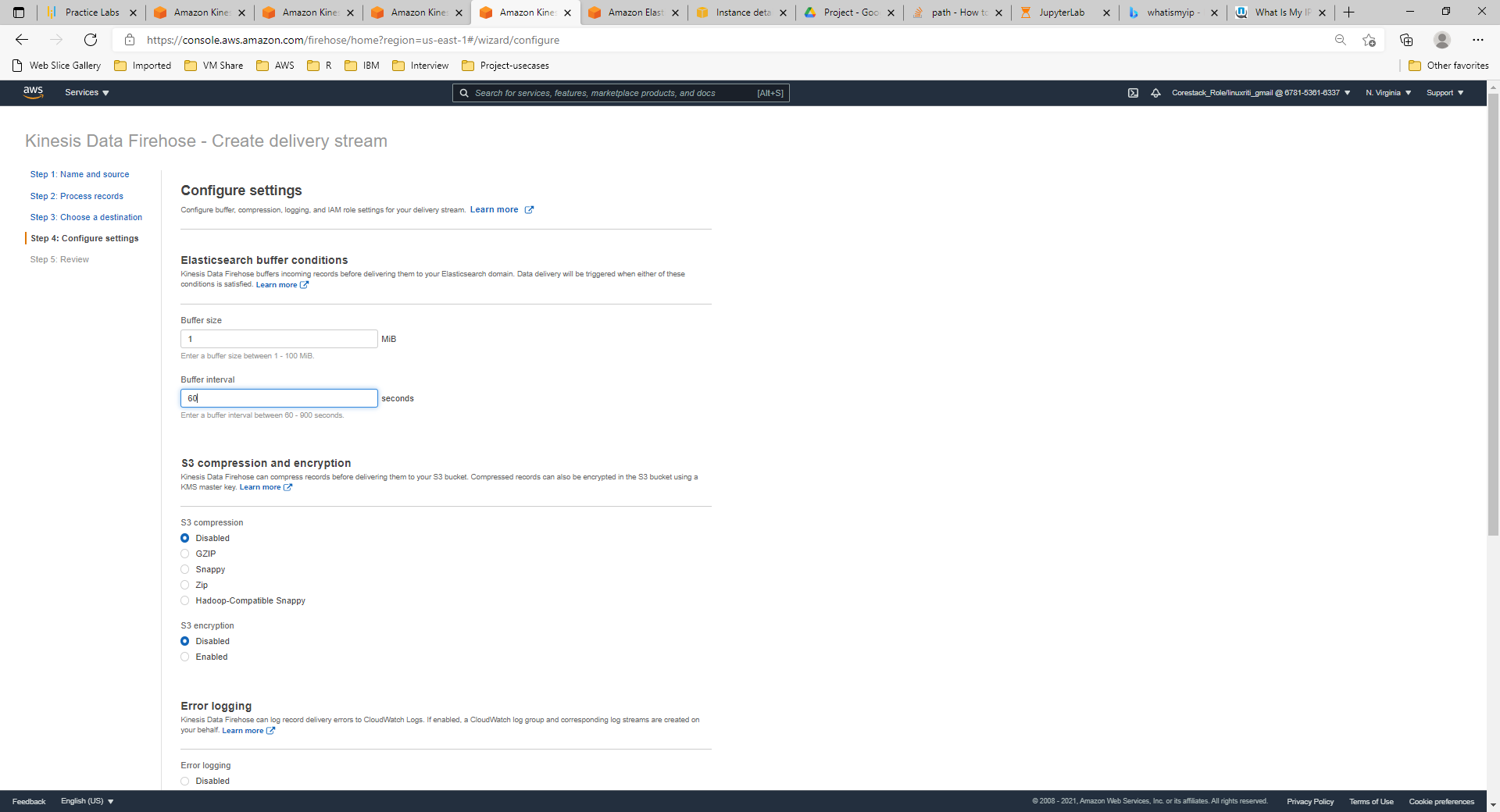


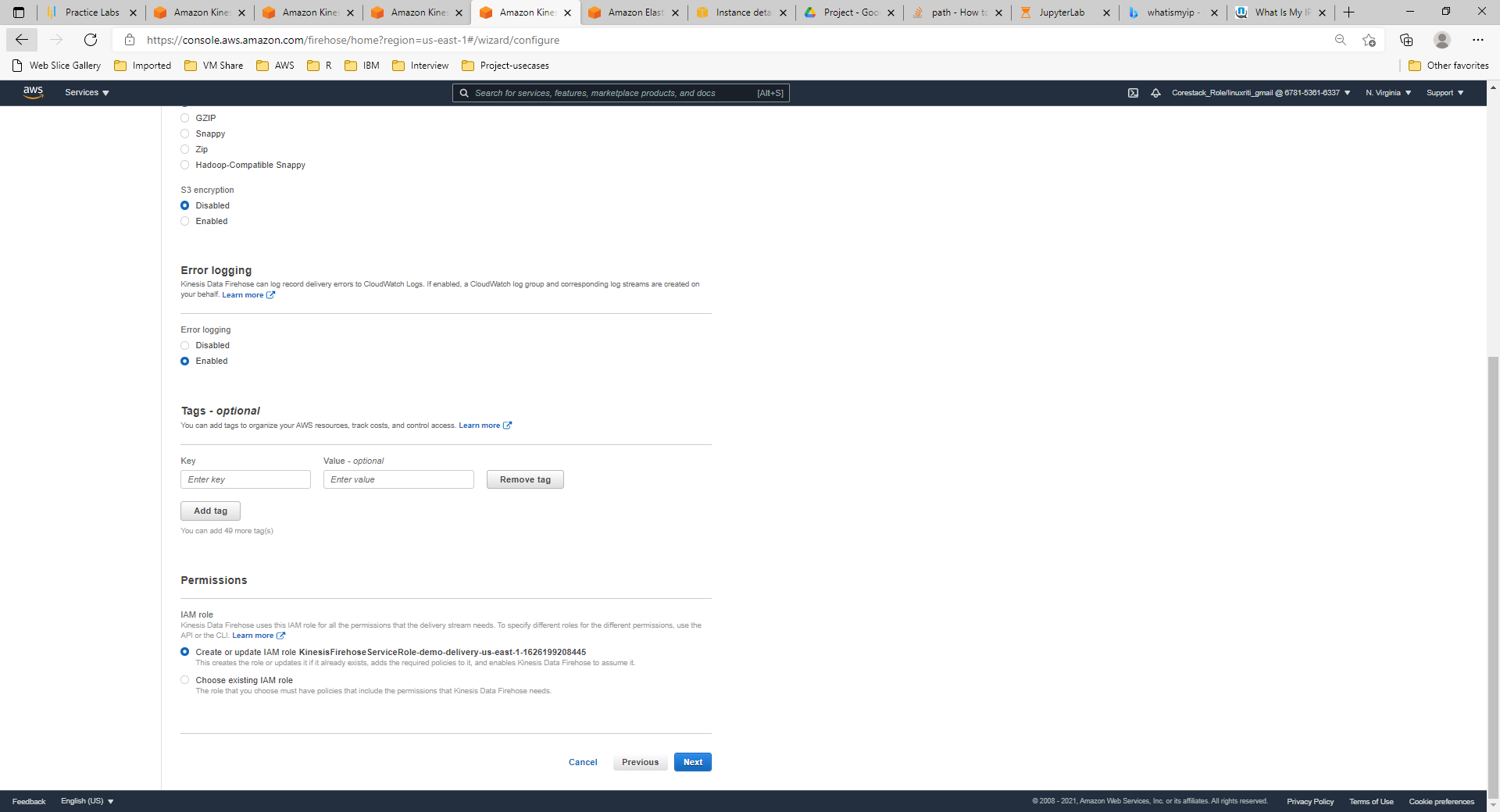


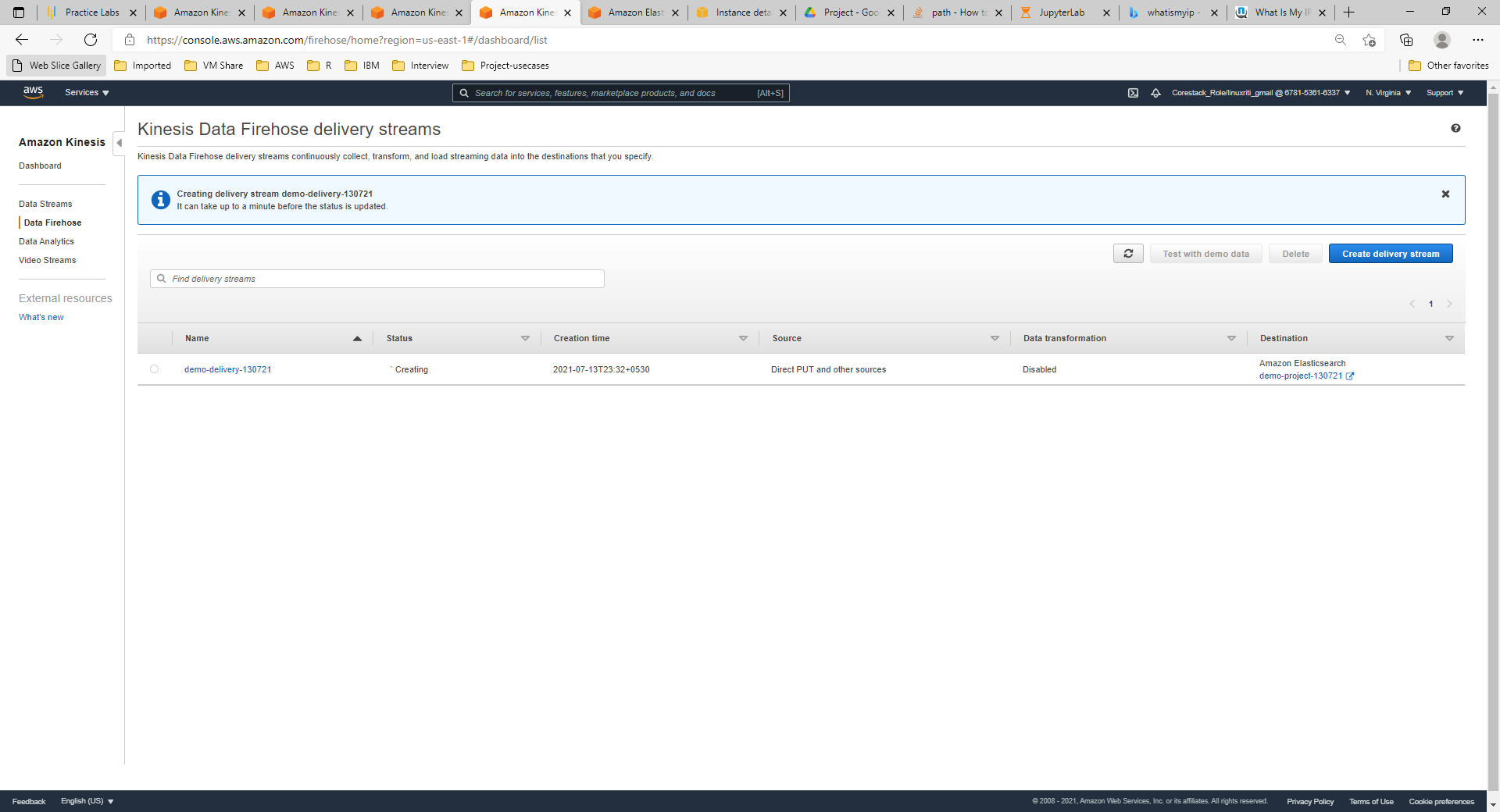




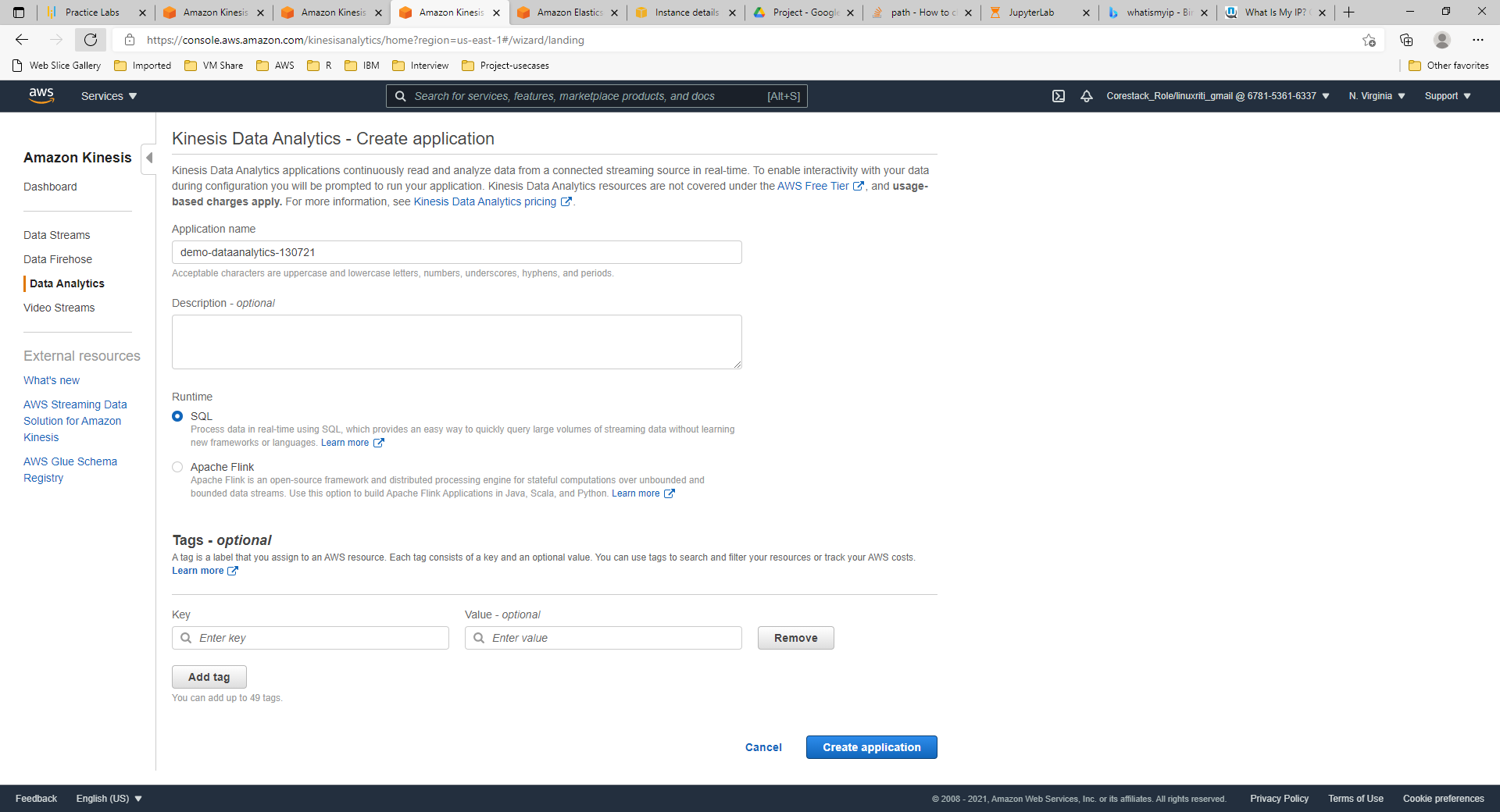
**Step 5:** Update the buffer size and existing IAM role for the process



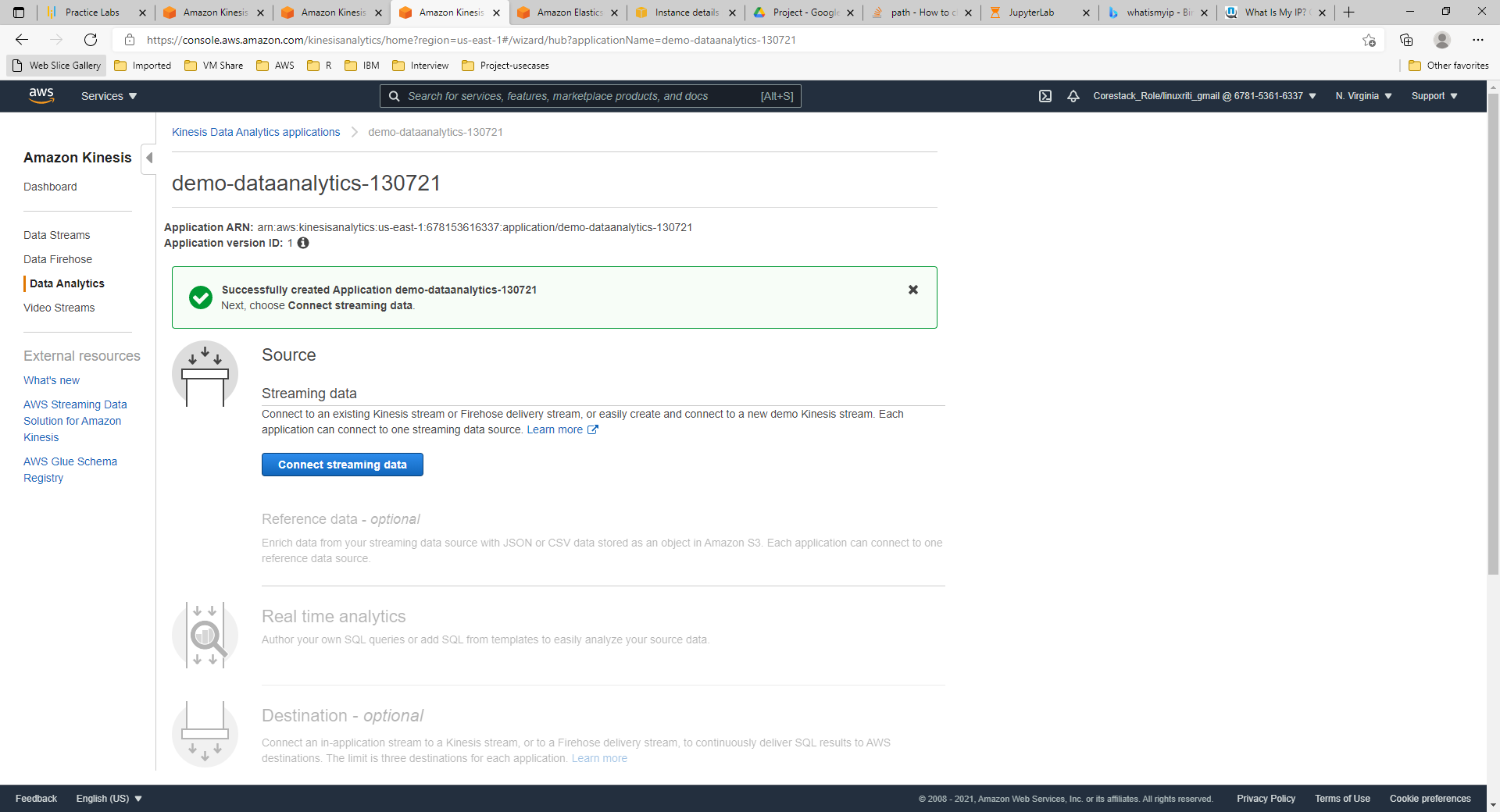


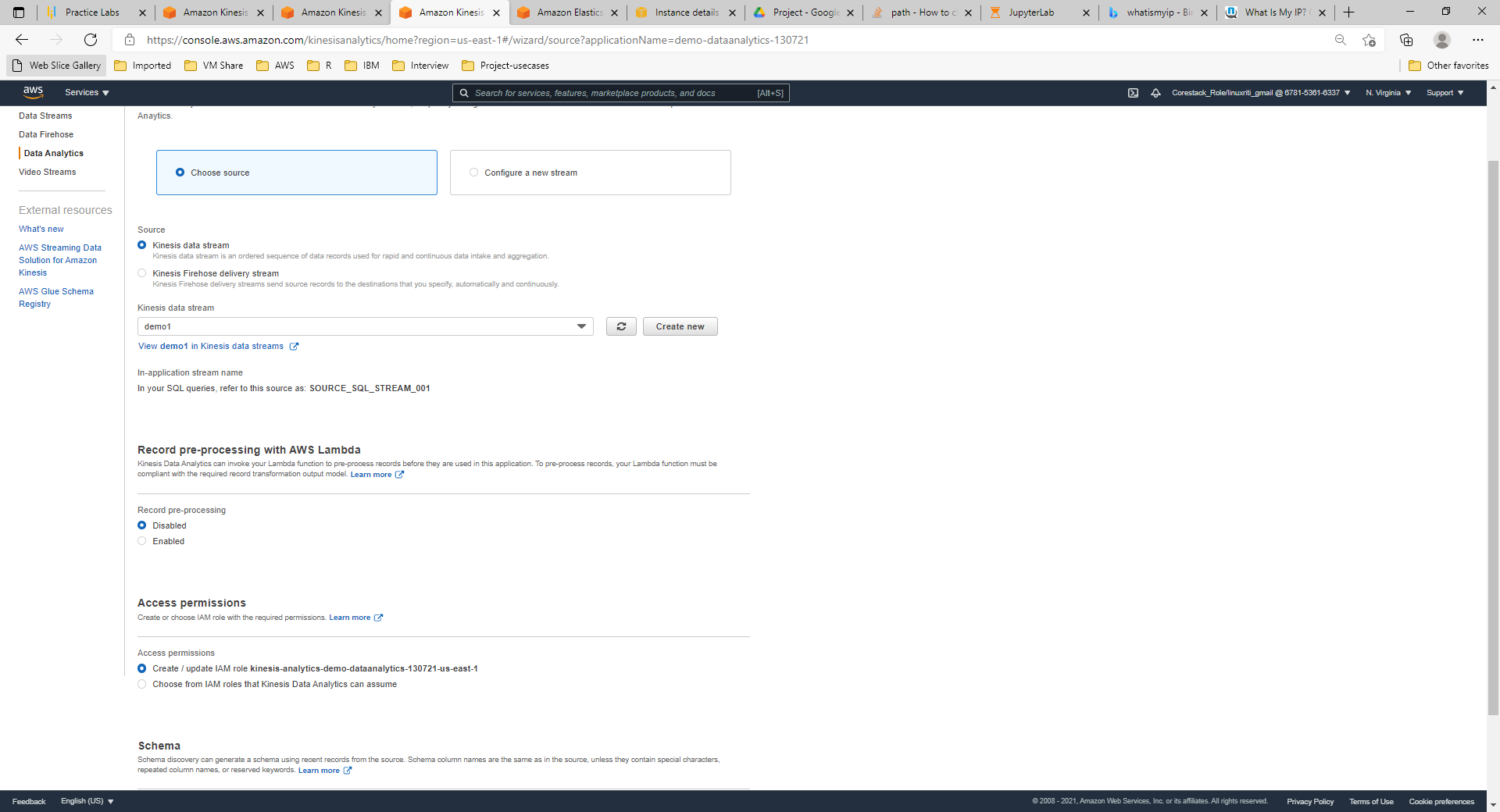


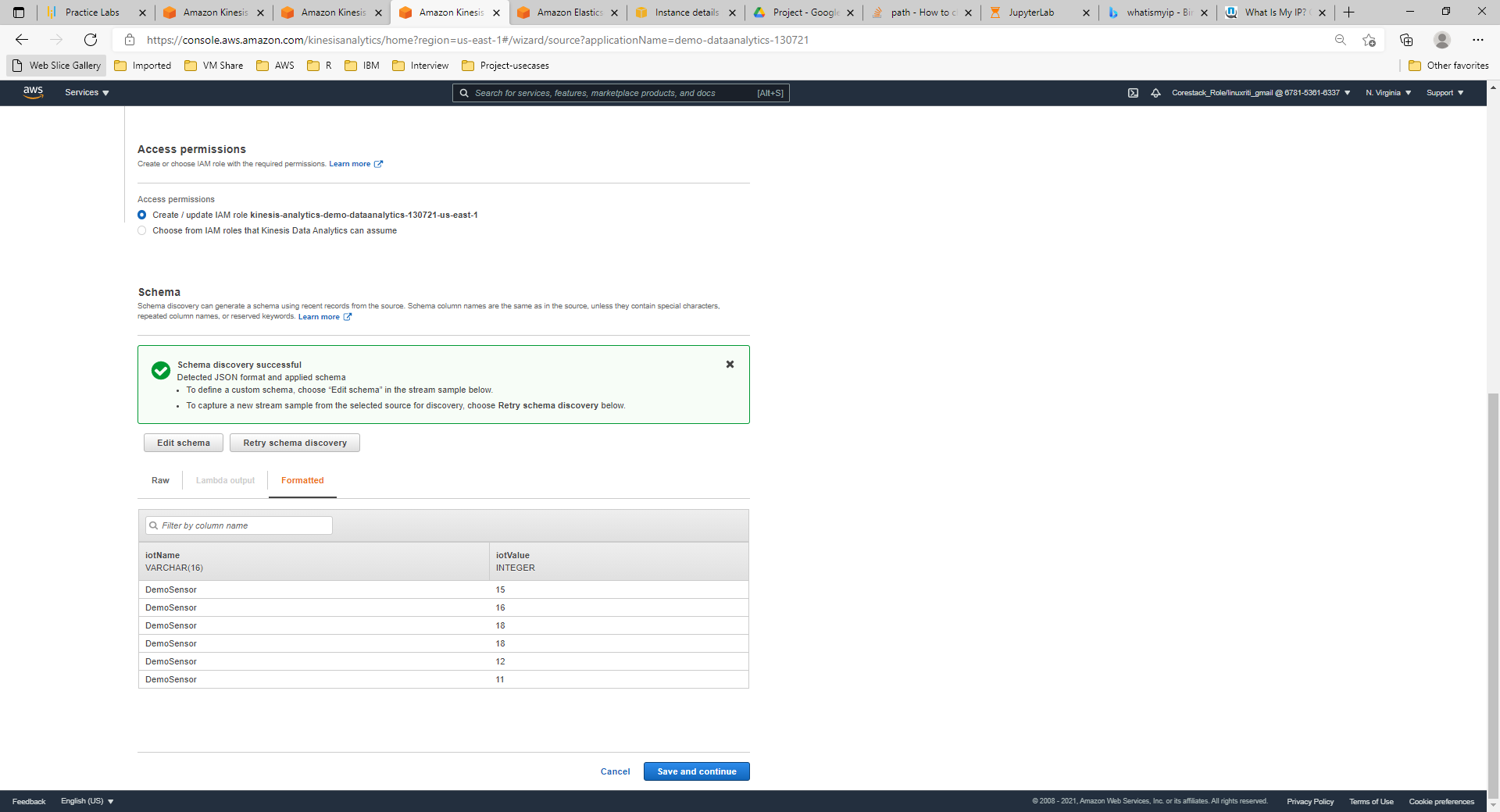
**Step 6: Open the Amazon Kinesis Analytics console and create a new application**

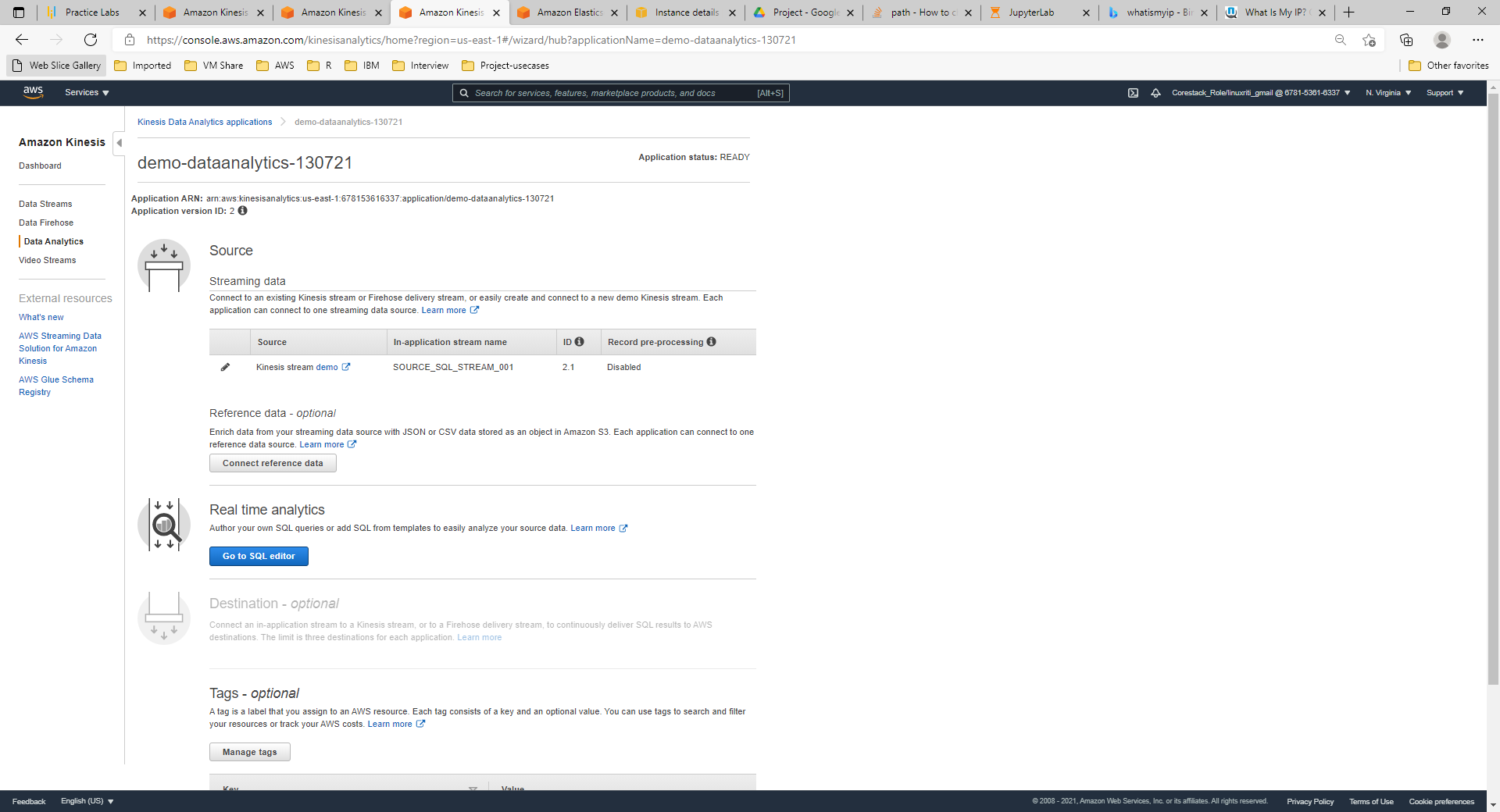


**Step 7:** Connect to the source for further analysis

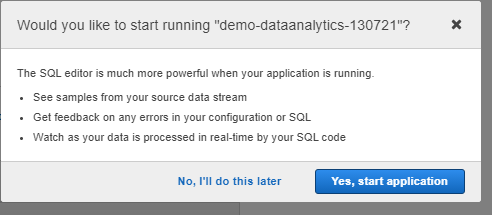


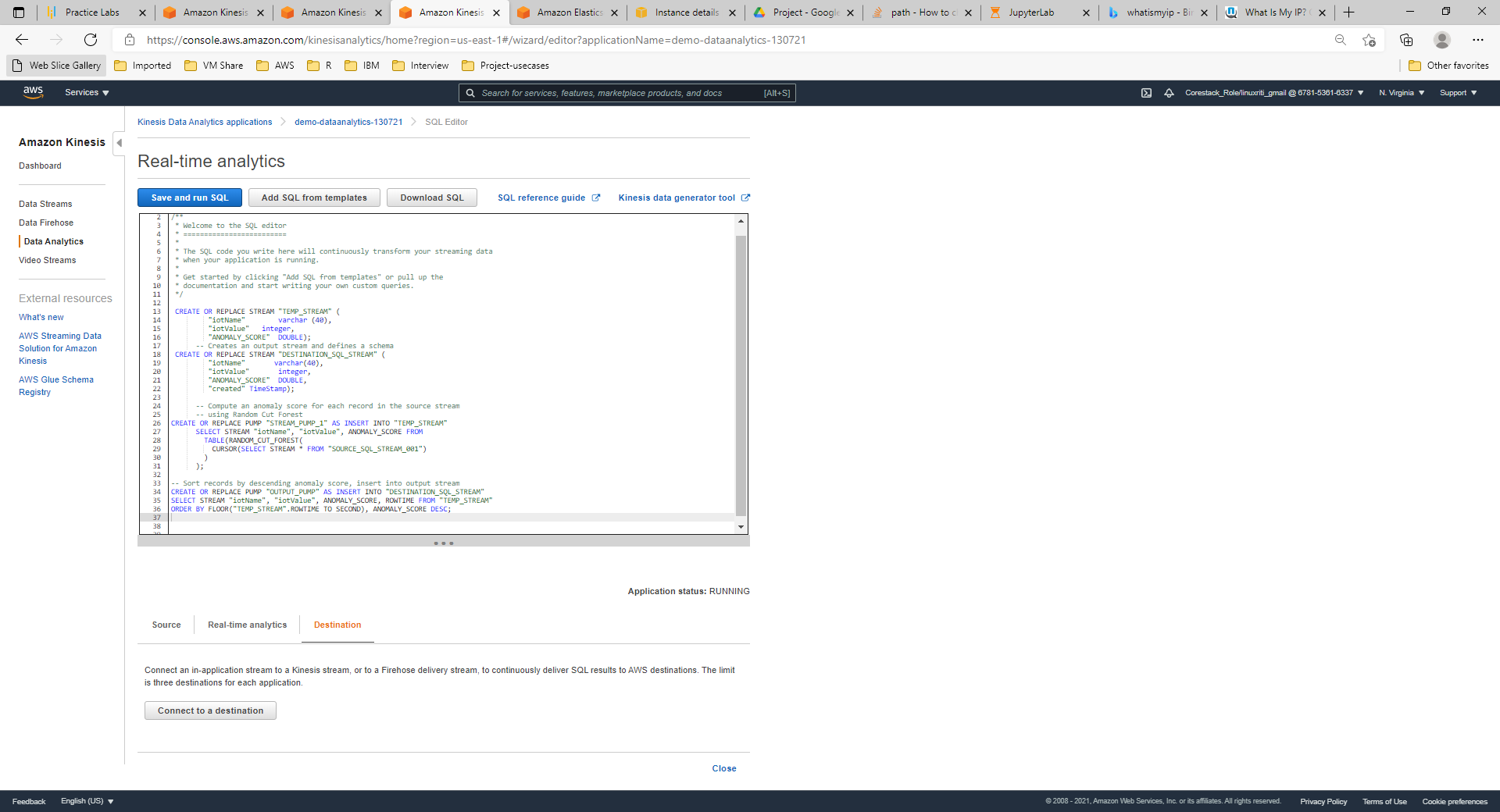


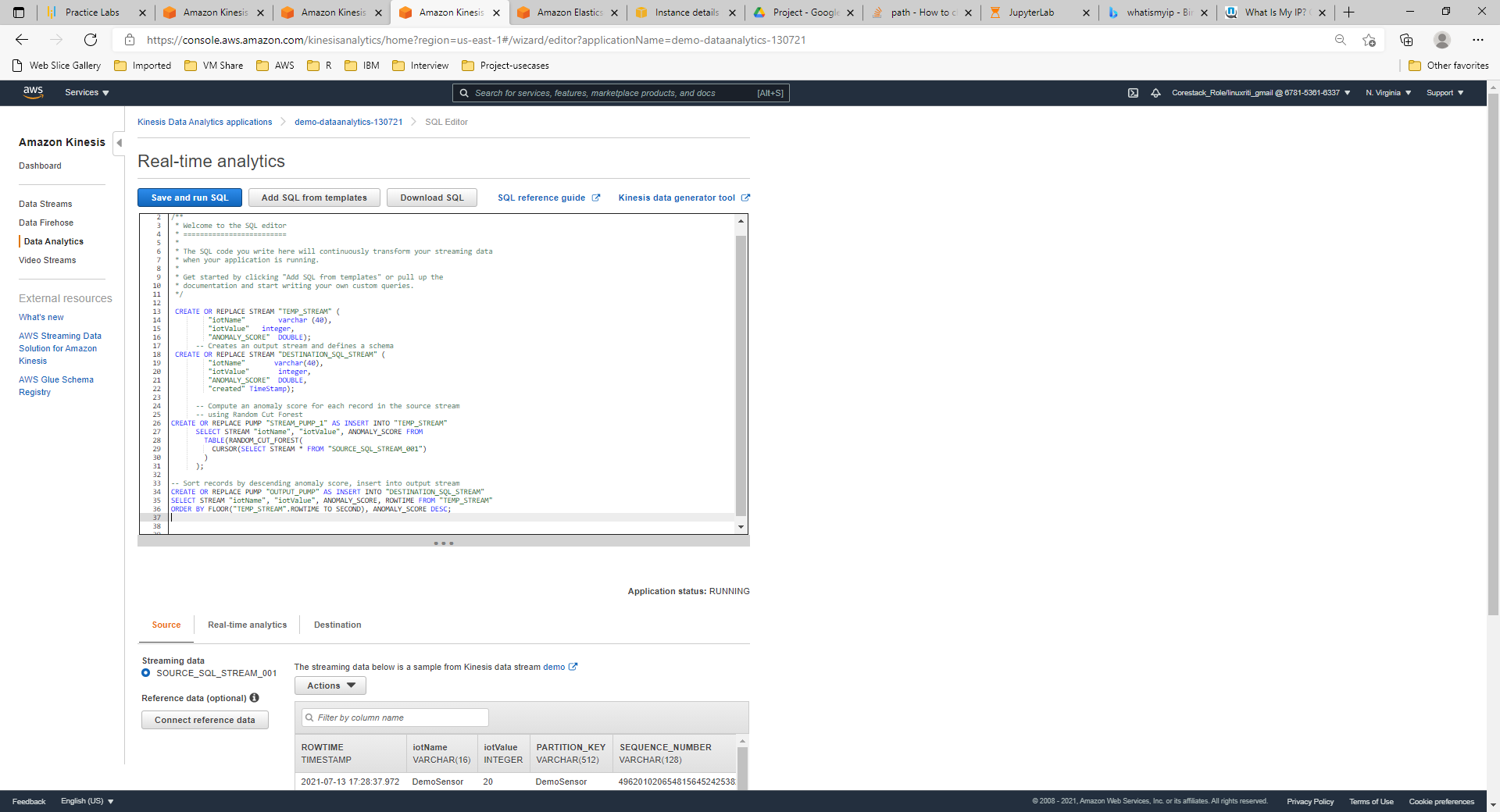


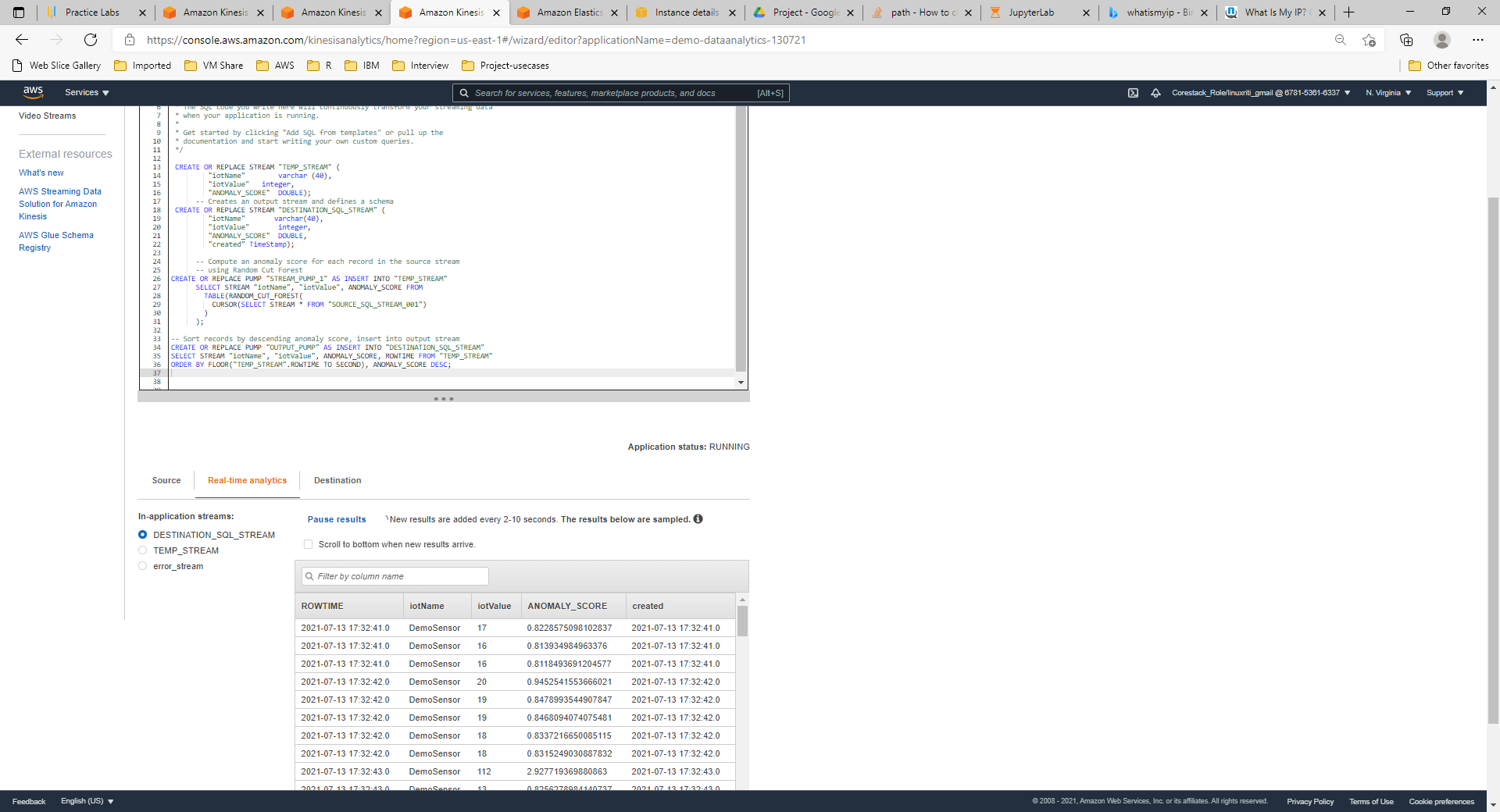


**Step 8:** Launch SQL\_Editor and start the application

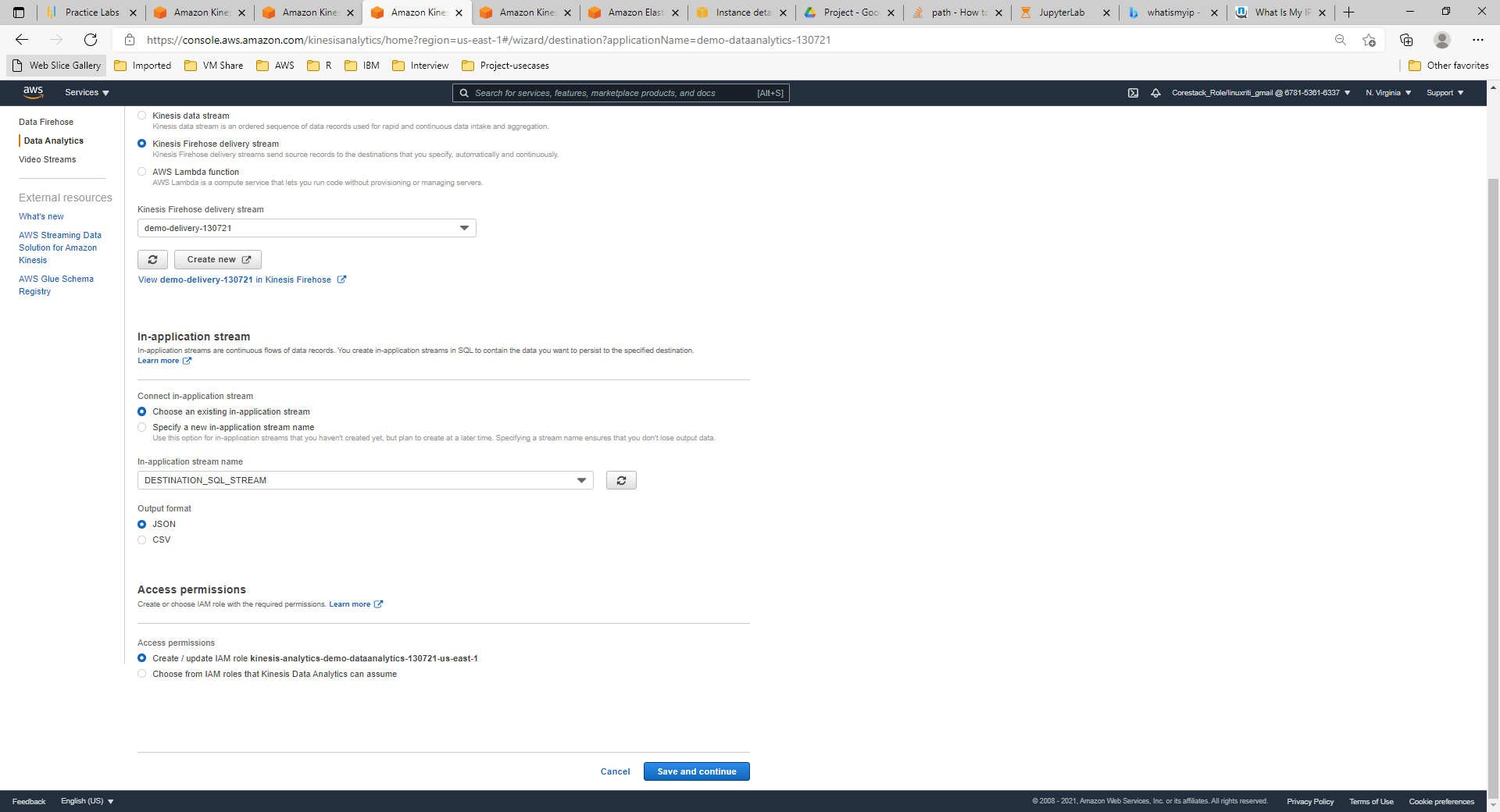


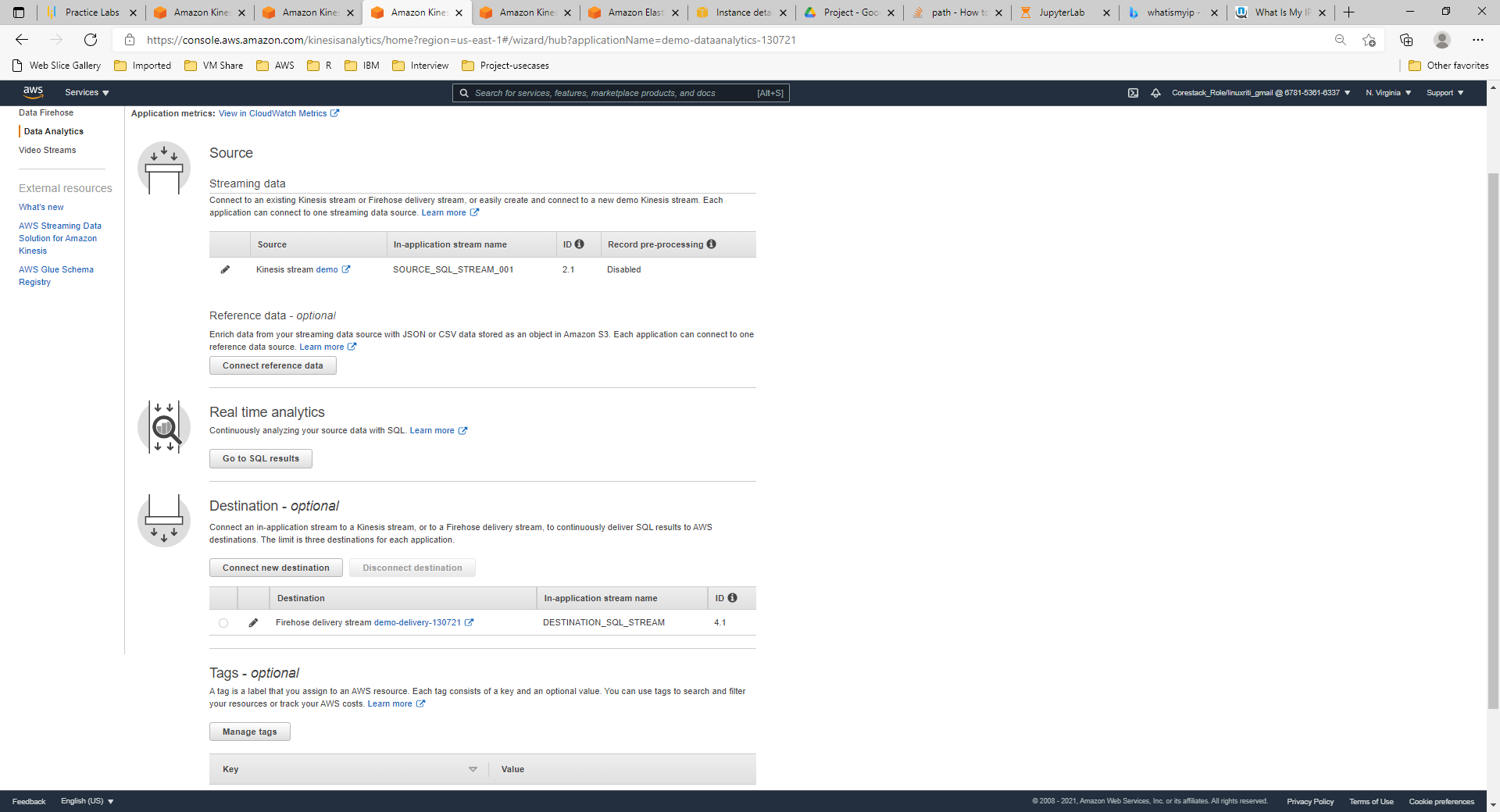


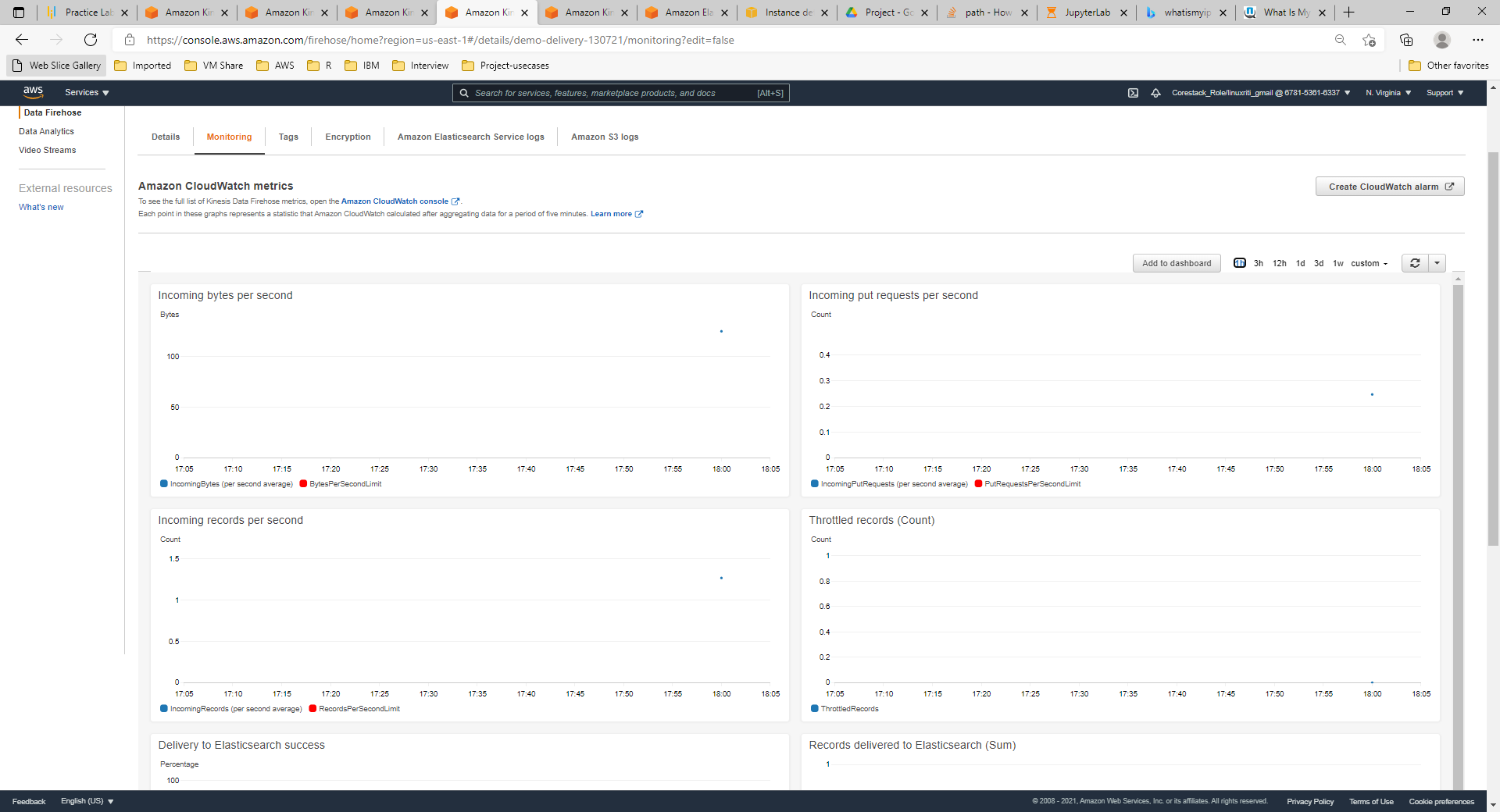


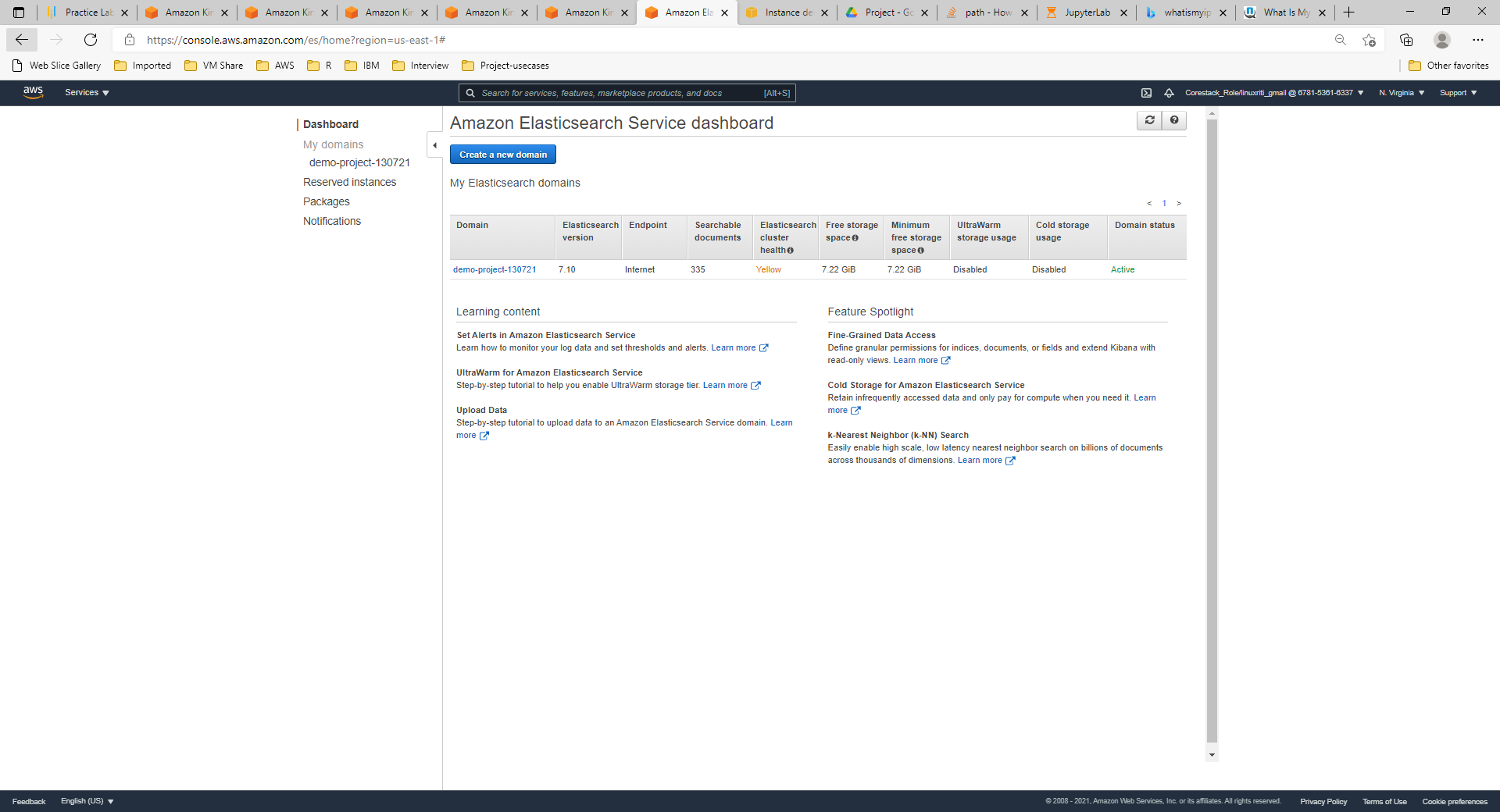


Step 9: Load the processed data into Kinesis Firehose delivery stream









Step 10: Visualize the data using Kibana

