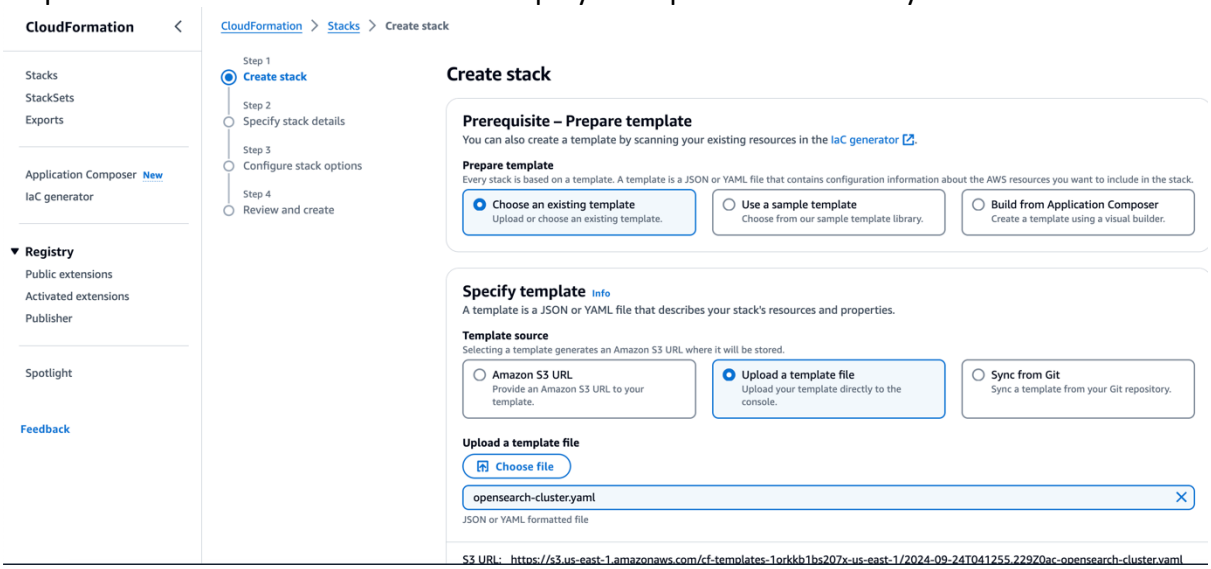
Solution: Amazon Connect real-time analysis with Opensearch Service (Cluster)

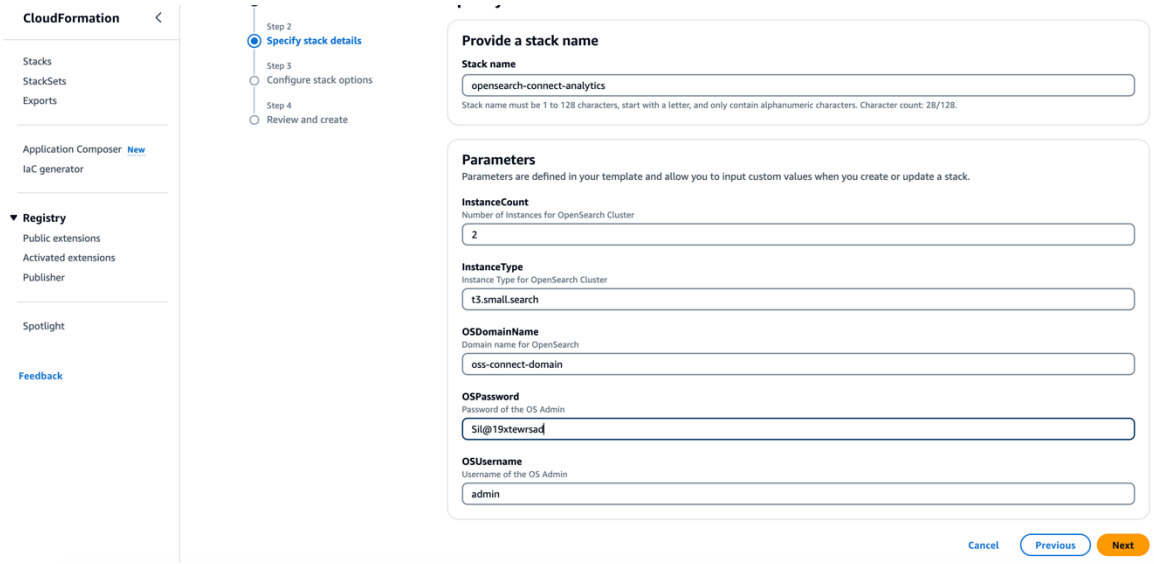
1. Setup Opensearch service

   Provision an Opensearch Cluster using the below yaml script.
   https://github.com/Fraser27/agentic-rag/blob/main/opensearch-cluster.yaml

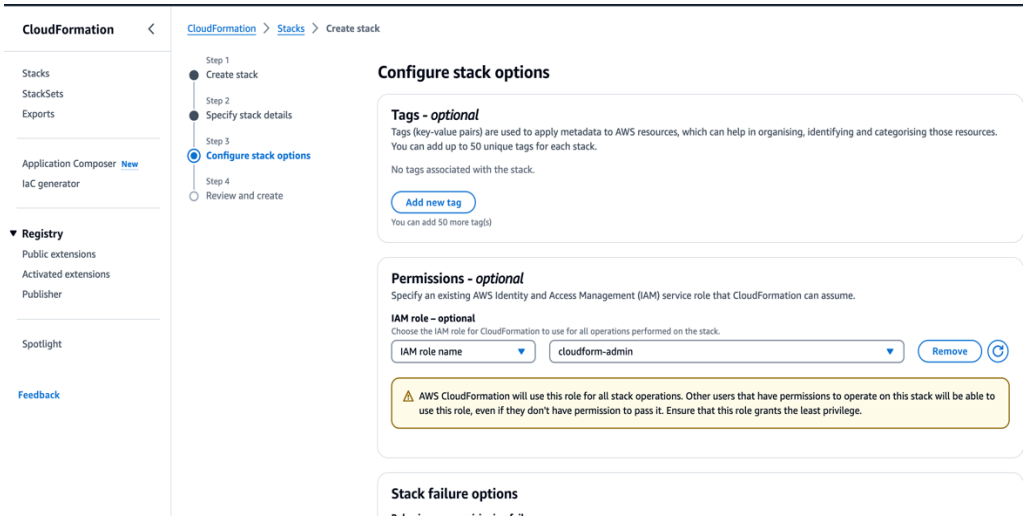
   Step 1.1: Head to Cloudformation and deploy the Opensearch-cluster.yaml file



   Step 1.2: Do not change the instance count. You can change the data node types as per your usecase. For production use-cases type in a different node-type, options are available here

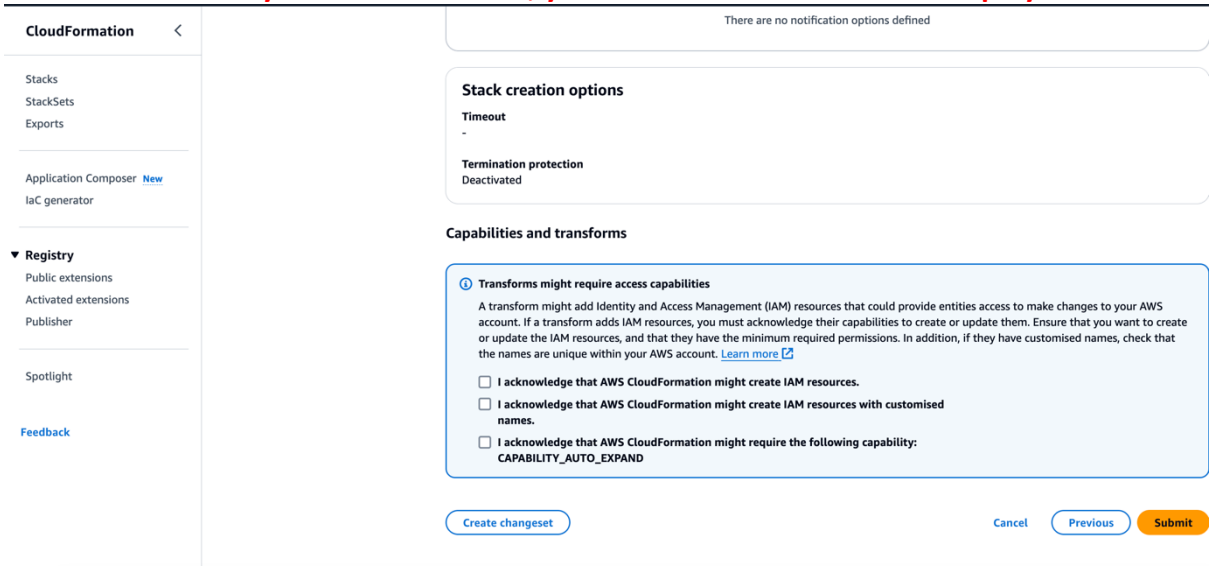


Step 1.3: On the Next page you should select an IAM role that has full ADMIN access on your AWS account that allows cloudformation to create the resources on your behalf

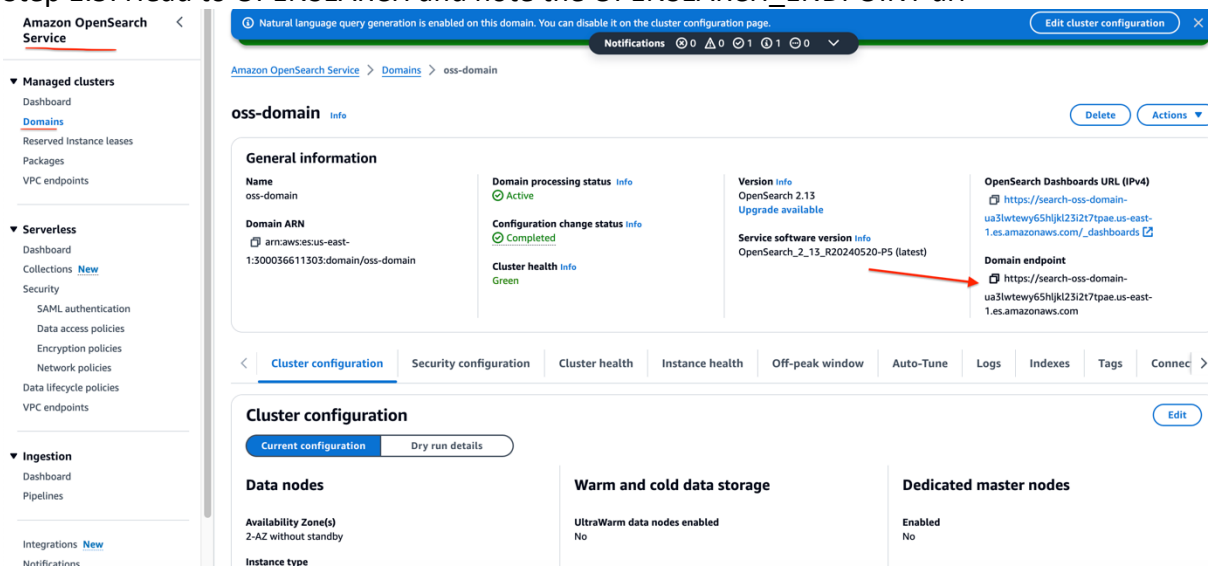


Step 1.4: Tick these checkboxes on the next page and hit **Submit. The stack takes roughly 20 minutes to deploy based on instance count.**

Note: Do not modify the instance count, you can do it once the stack is deployed

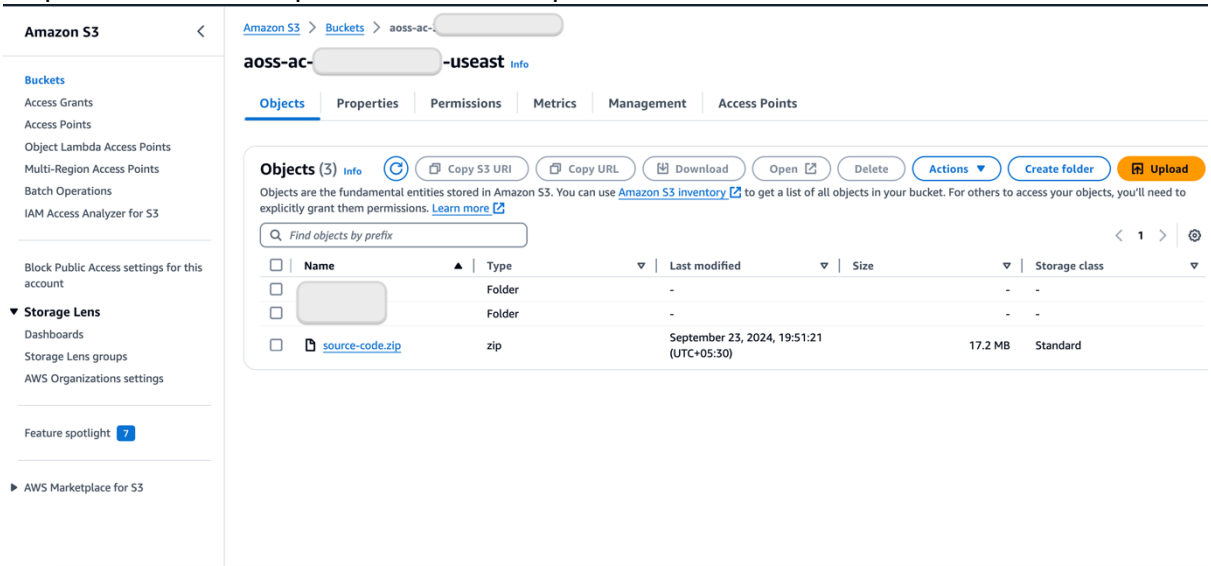


Step 1.5: Head to OPENSEARCH and note the OPENSEARCH_ENDPOINT url



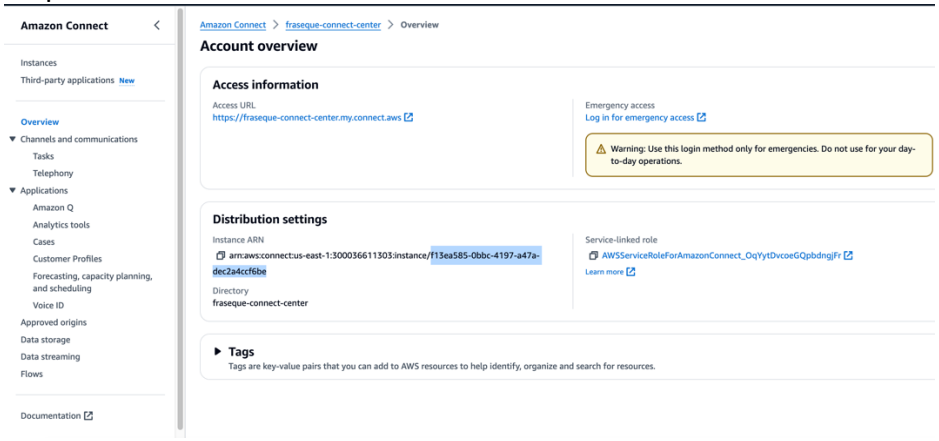
2. Create an integration pipeline between Amazon Connect and Opensearch

Step 2.1: Create an S3 private bucket with prefix aoss- as follows

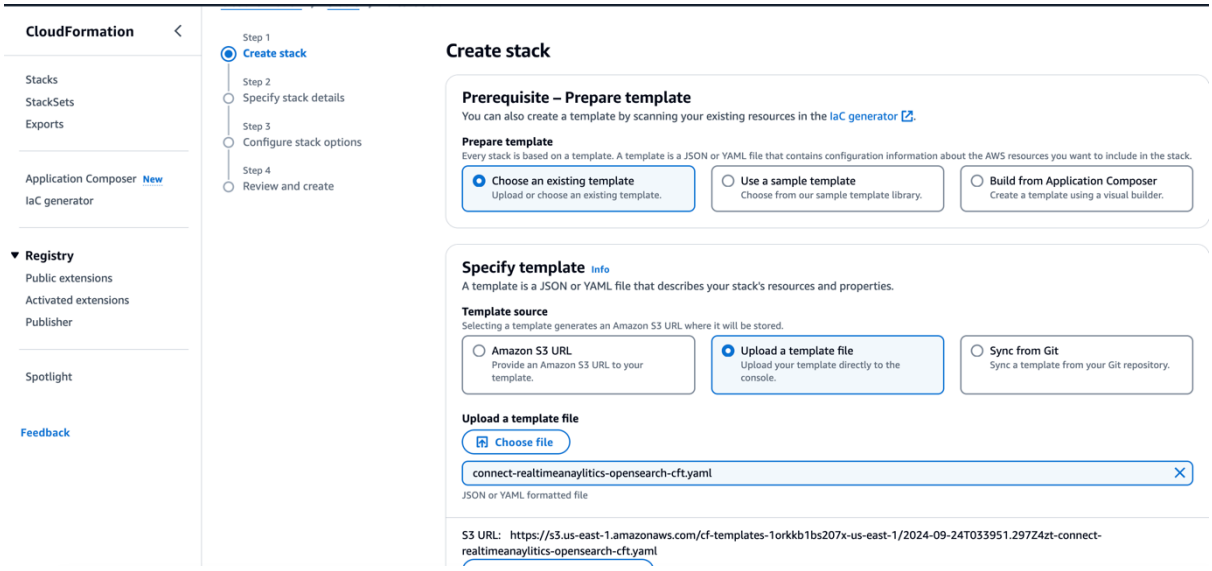


Step 2.2: Drop in the source-code.zip into the newly created S3 bucket, It's available in this github repository.
https://github.com/Fraser27/amazon-connect-analytics/blob/main/source-code.zip

Step 2.3: Fetch the Amazon Connect ID

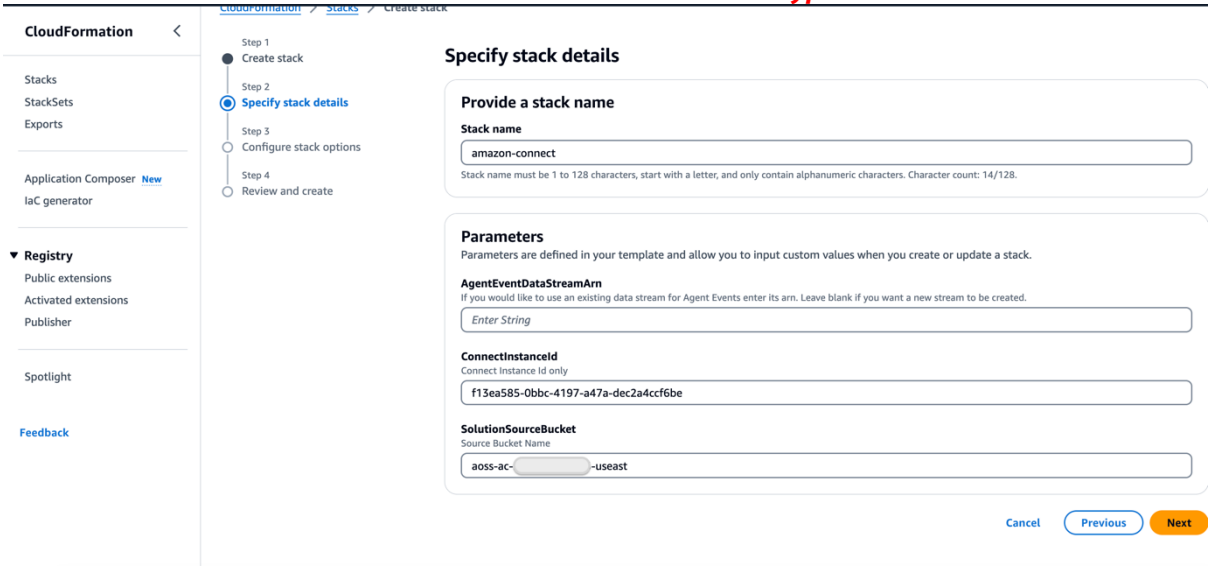


Step 2.4: Go to Cloudformation, download and deploy the YAML file from here https://github.com/Fraser27/amazon-connect-analytics/blob/main/connect-realtimeanaylitics-opensearch-cft.yaml

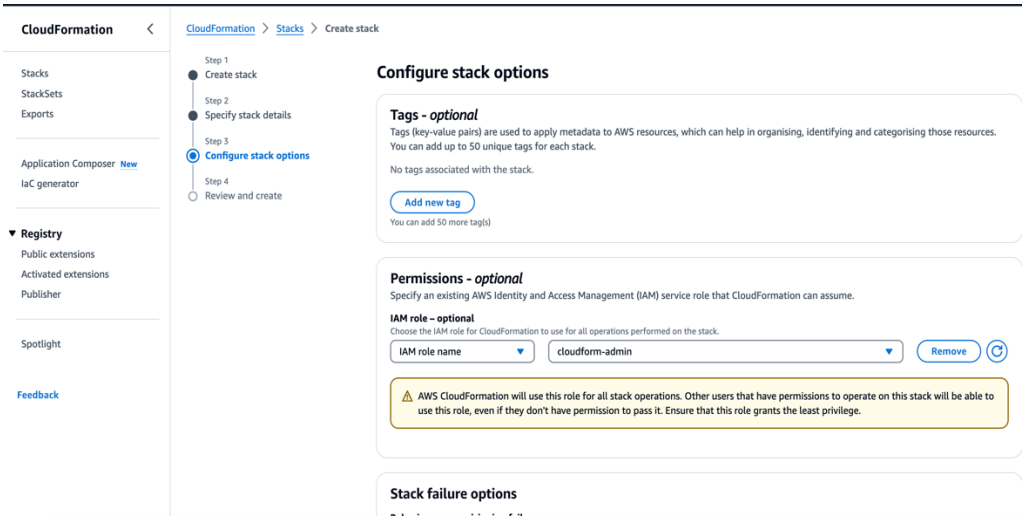


Step 2.5: Fill in the stack parameters as follows. You may leave the **AgentEventDataStreamArn** as blank so a new Kinesis stream is created for you. Hit **Next**

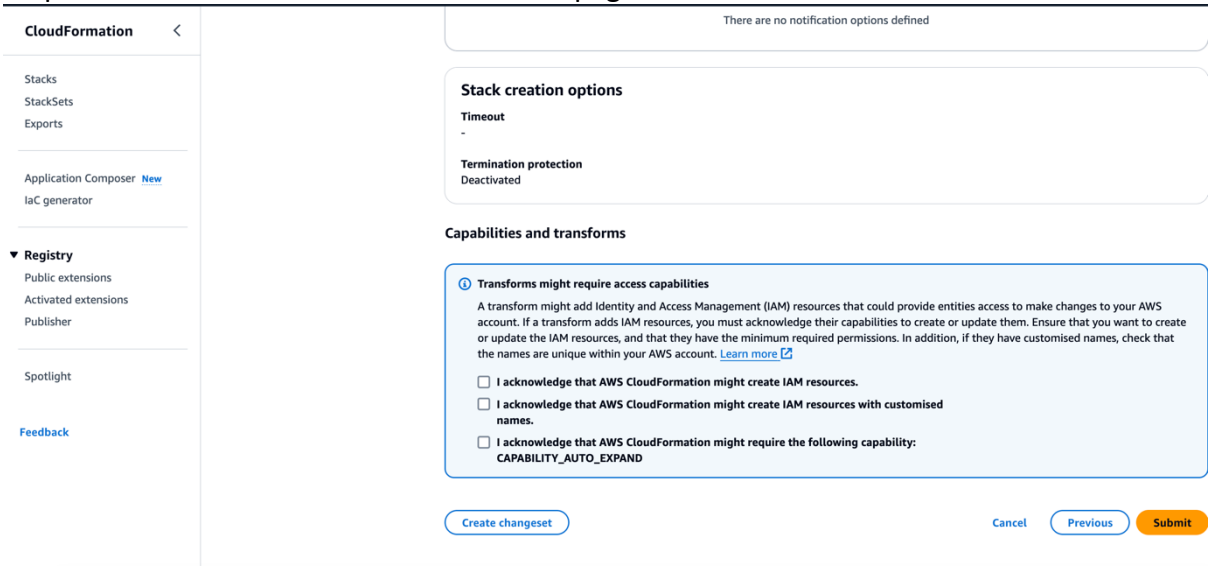
Note: ***Stack name should all be lowercase and can contain hyphens or numbers***



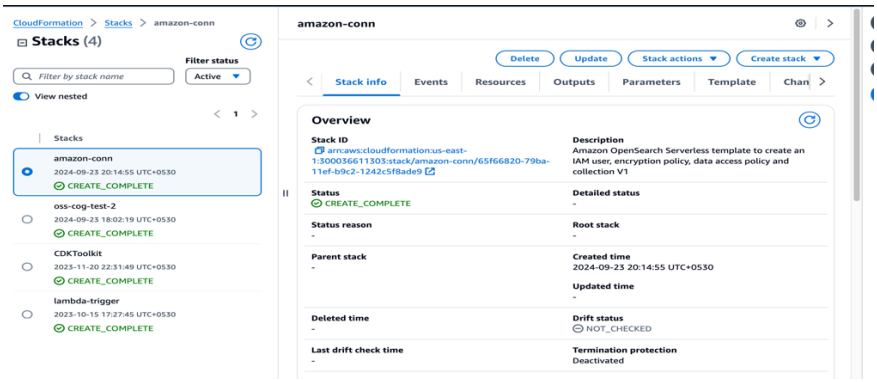
Step 2.6: On the Next page you should select an IAM role that has full ADMIN access on your AWS account that allows cloudformation to create the resources on your behalf



Step 2.7: Tick these checkboxes on the next page and hit **Submit**

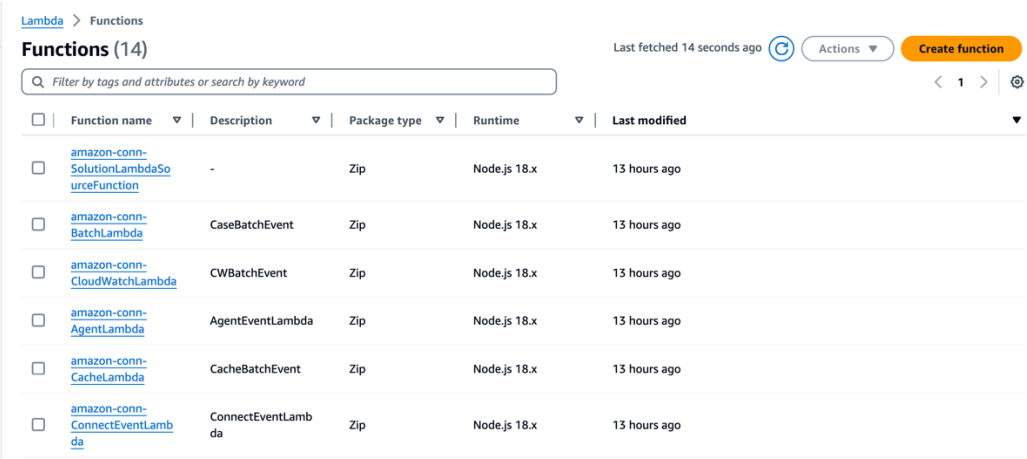


Step 2.8: Wait for the stack to finish deployment

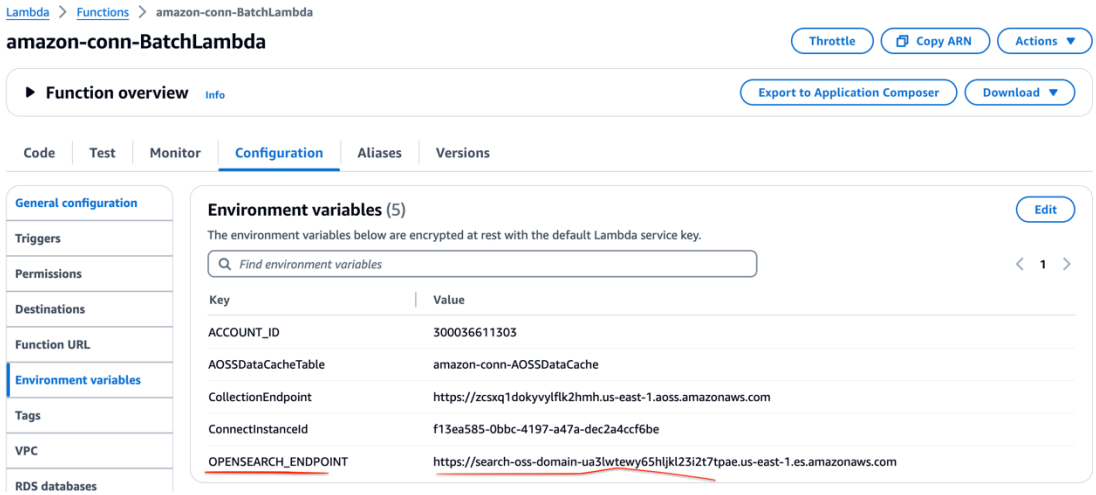


3. <u>Add the Opensearch Endpoint URL as an environment variable to your lambda functions.</u>

Step 3.1: Below are the 6 lambdas created by your cloudformation script for which we now will add the Opensearch Endpoint environment variable

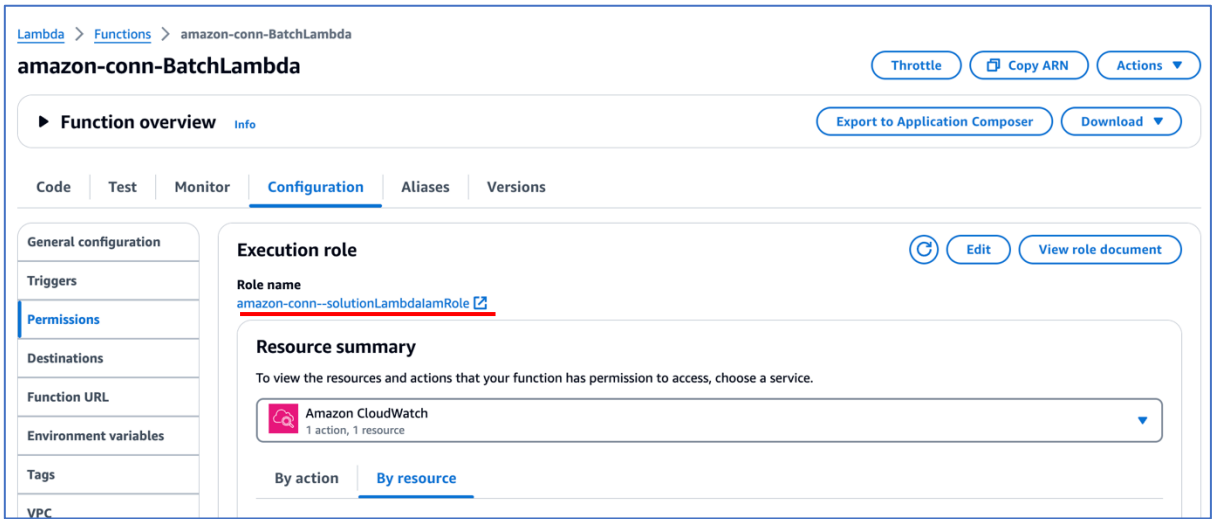


Step 3.2: Do the below exercise for all 6 lambdas. Create OPENSEARCH_ENDPOINT environment variable for your lambda function and point it to the Opensearch DOMAIN URL identified in Step1.

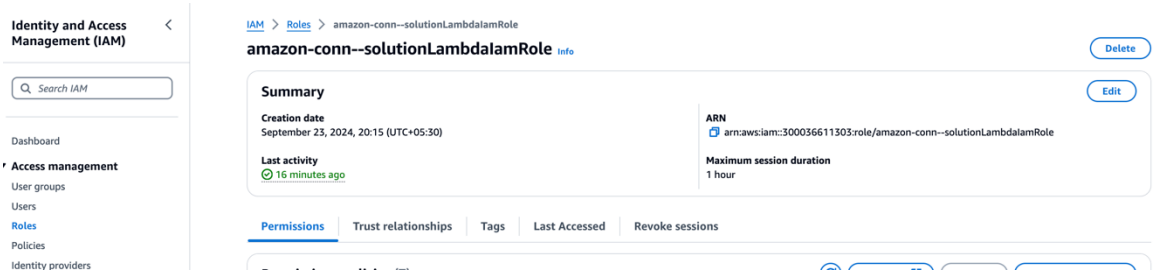


4. <u>Add a Backend role to allow Lambda to access Opensearch</u>

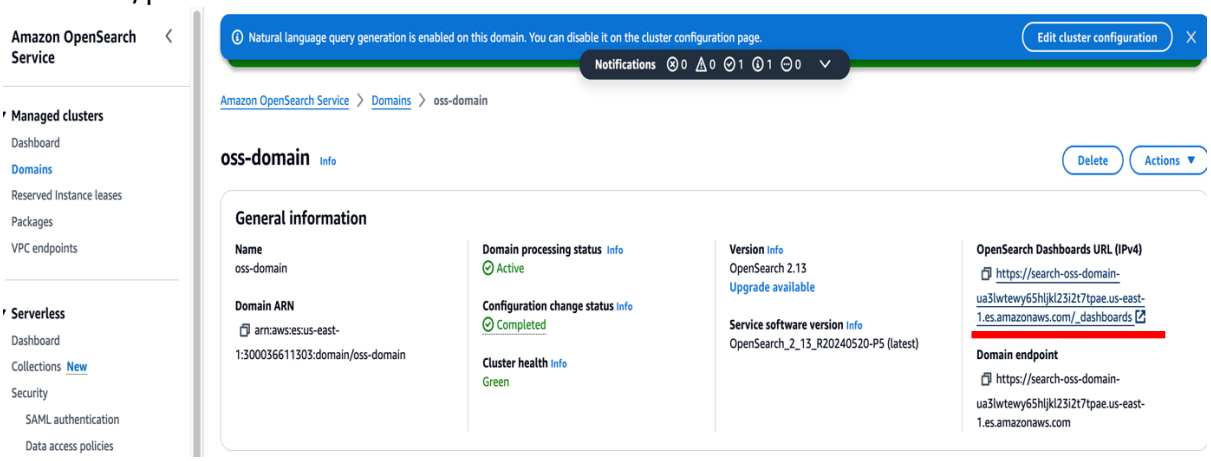
Step 4.1: Identify the LAMBDA role. Click on any lambda function and head to configuration.



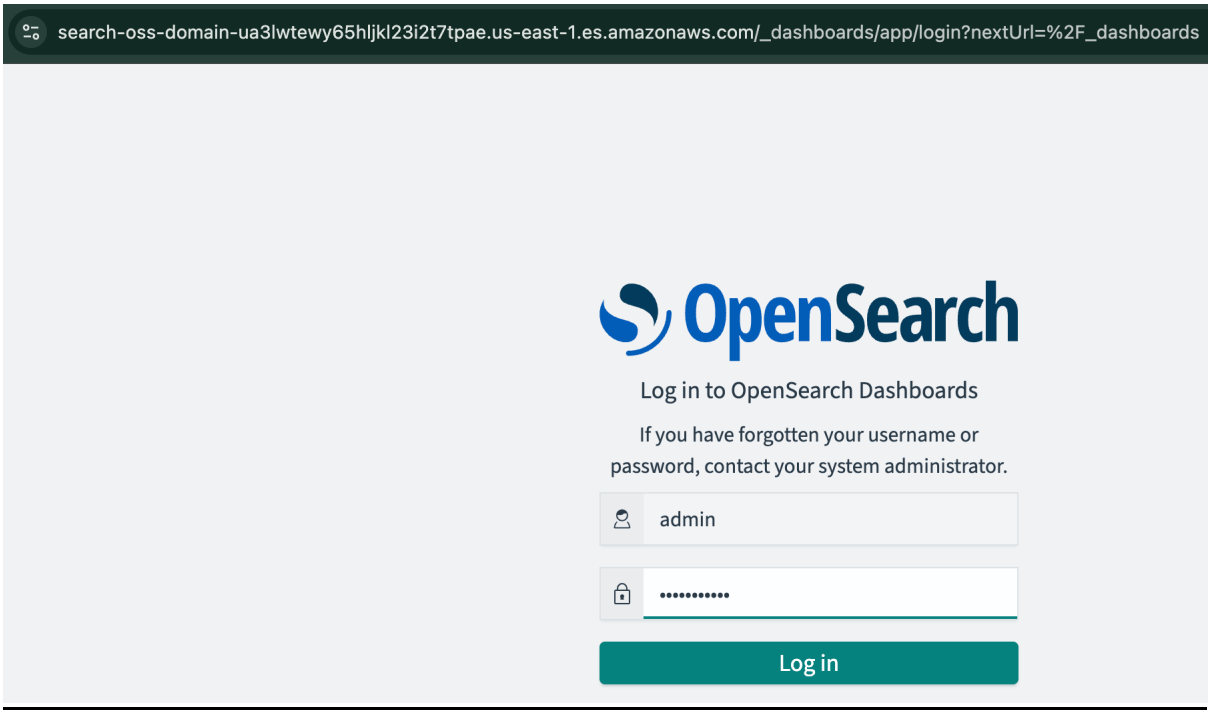
Step 4.2: Head to IAM search for the lambda role and Identify the ARN of the lambda role as below



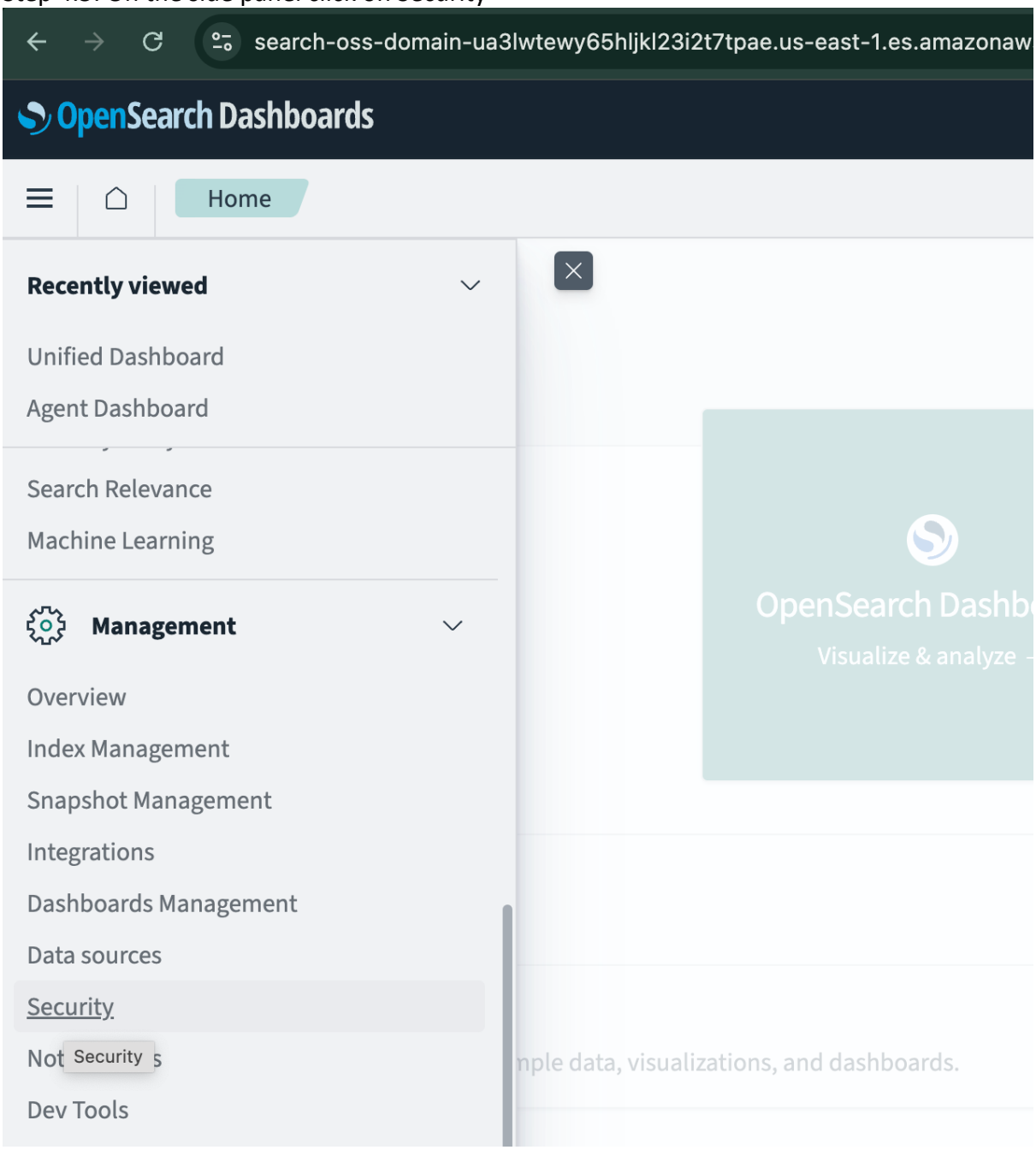
Step 4.3: Head to Opensearch and click on the dashboard URL. Login with Master username/password



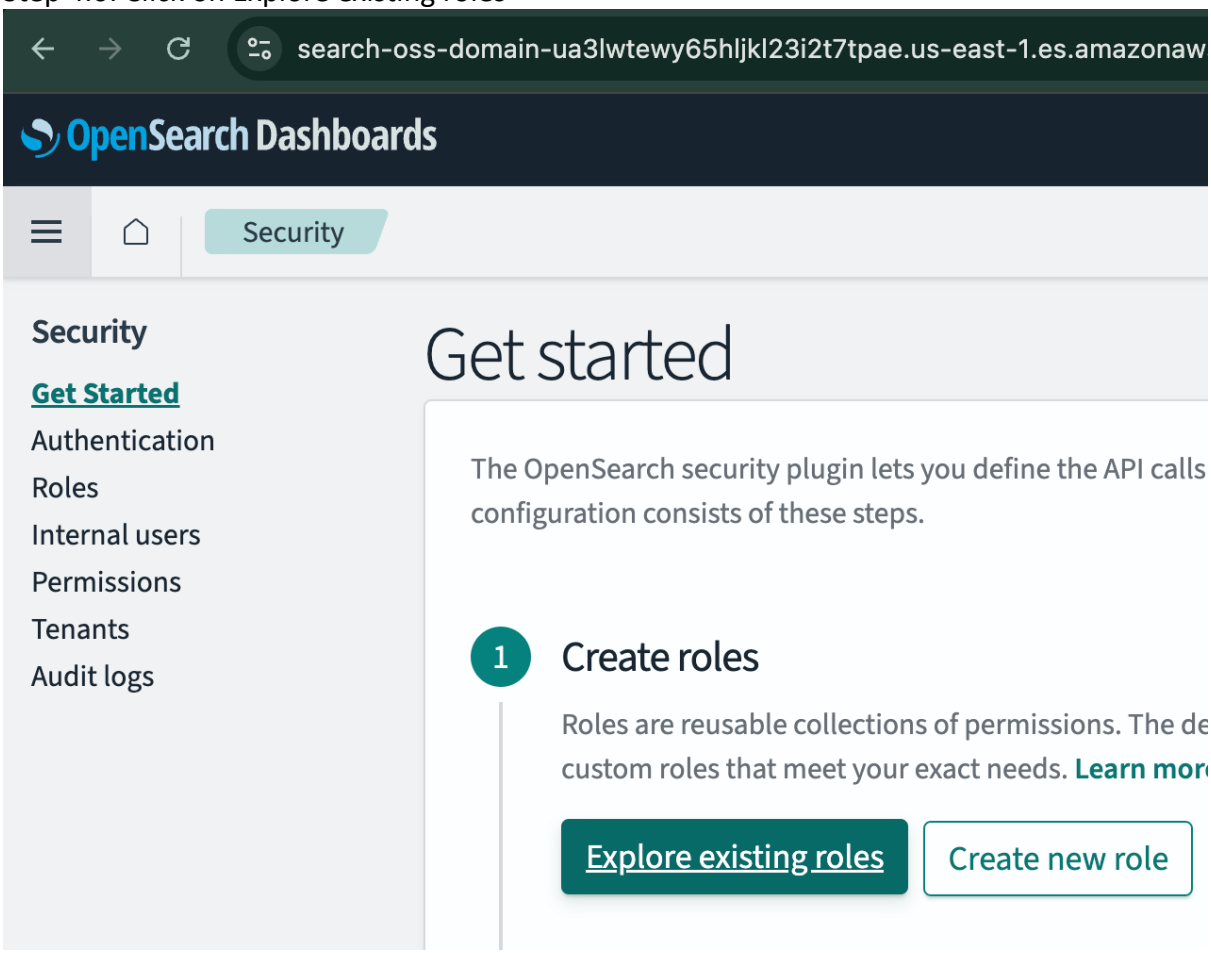
Step 4.4: Login to Opensearch with Master Username/Password


search-oss-domain-ua3lwtewy65hljkl23i2t7tpae.us-east-1.es.amazonaws.com/_dashboards/app/login?nextUrl=%2F_dashboards
OpenSearch
Log in to OpenSearch Dashboards
If you have forgotten your username or password, contact your system administrator.
admin
Log in

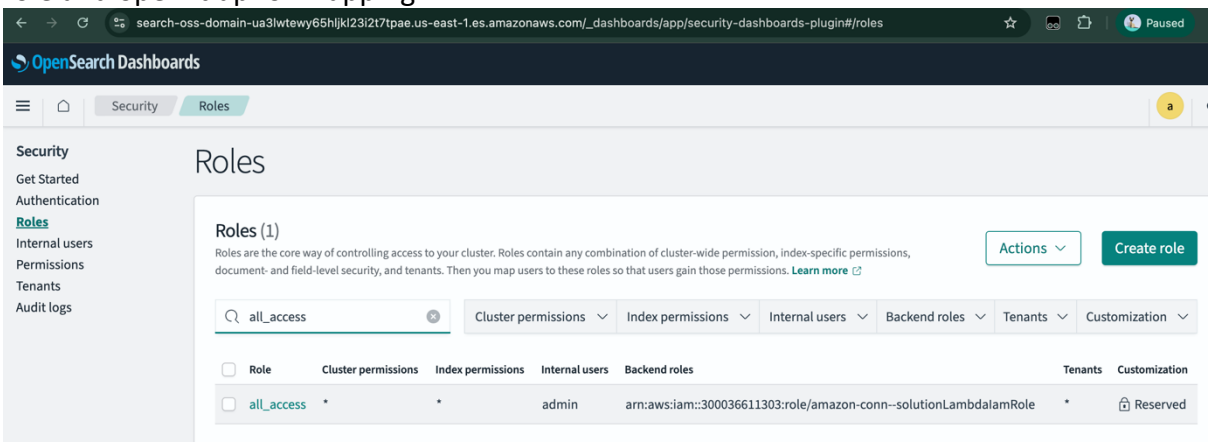
Step 4.5: On the side panel click on Security



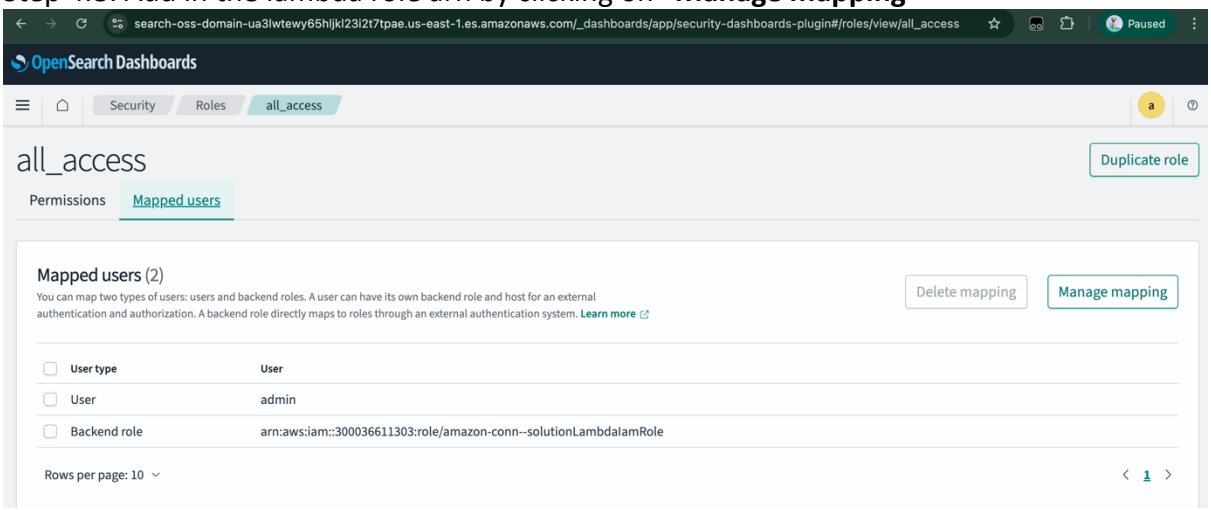
Step 4.6: Click on Explore existing roles



Step 4.7: Search for role all_access and map the role to our lambda role. Click on the role and open it up for mapping.



Step 4.8: Add in the lambda role arn by clicking on “**Manage mapping”**

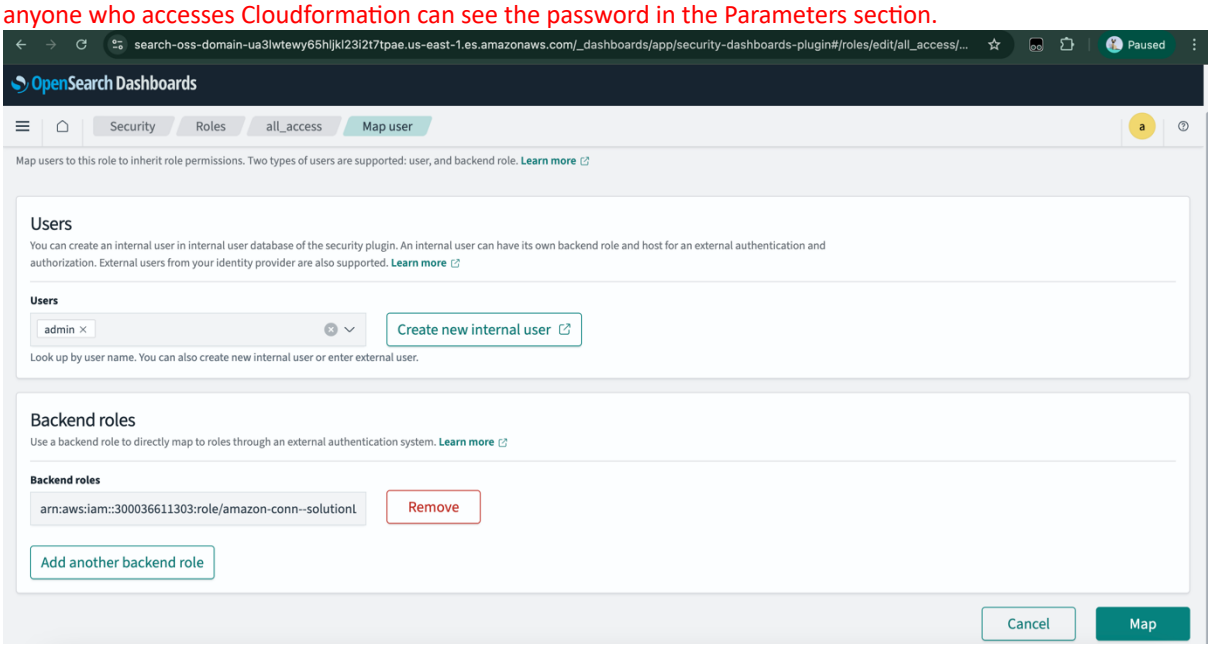


Step 4.9: Manage Mapping page looks as follows

Note: 1. We do not recommend adding in new users as it could create a security issue, anyone with username/password could access the system.

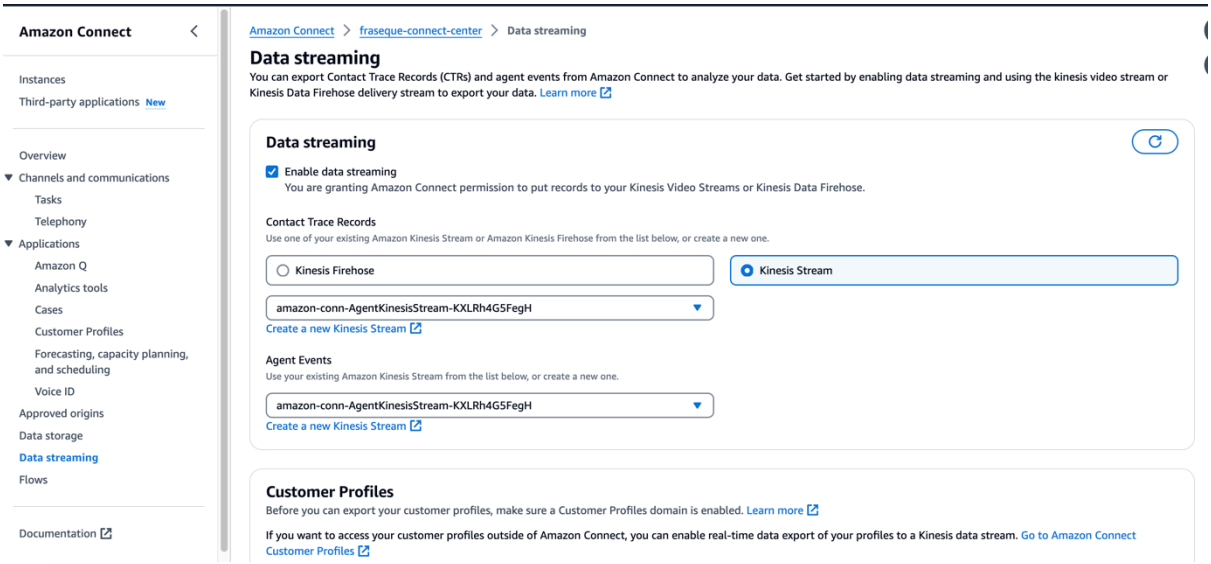
2. Instead create a role and only let a few users assume the role to access Opensearch. This page allows for fine-grained access. Here we are allowing the role on the lambda function to access Opensearch.

3. Also do modify the master username/password once the stack is deployed, this should be done as anyone who accesses Cloudformation can see the password in the Parameters section.



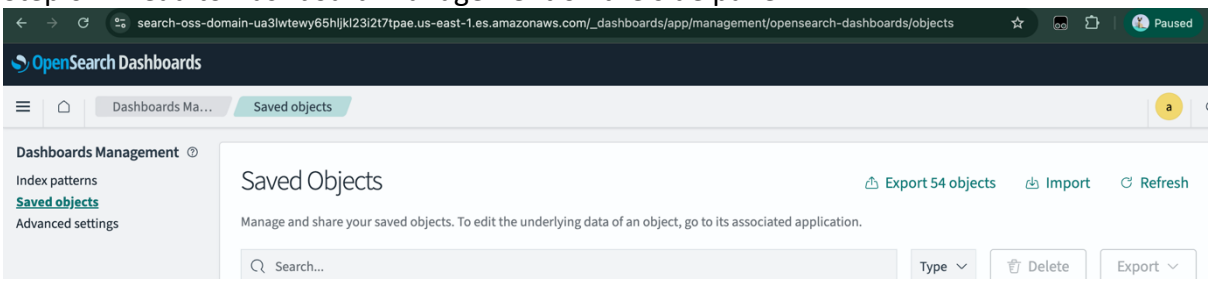
5. Enable Amazon Connect Data Streaming

Step 4.1: head to Amazon Connect on your AWS account. Enable streaming as shown in the below image. Select the newly created Kinesis Stream to push records to the lambda.

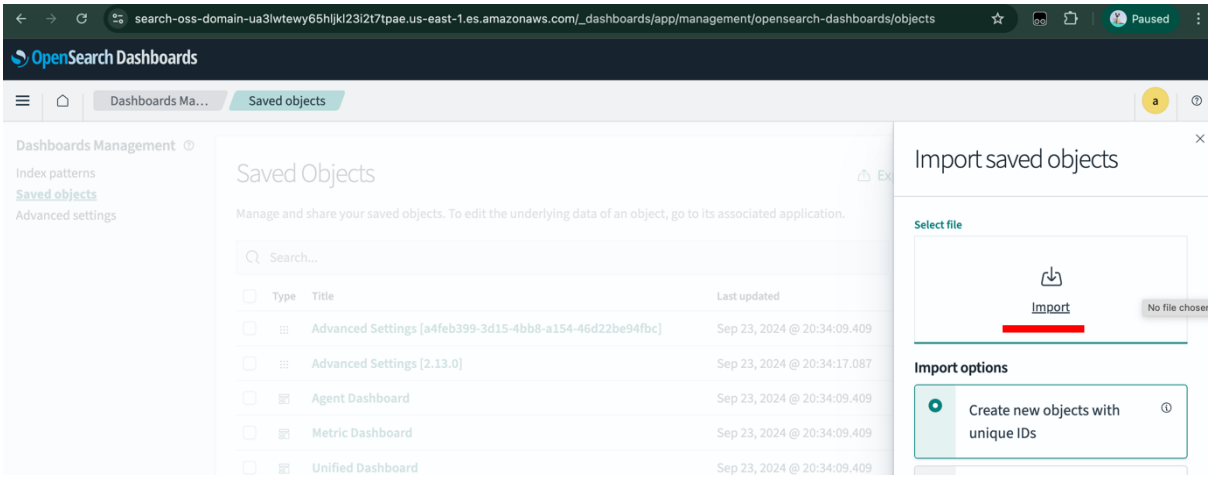


6. Import the dashboard objects into Opensearch as below

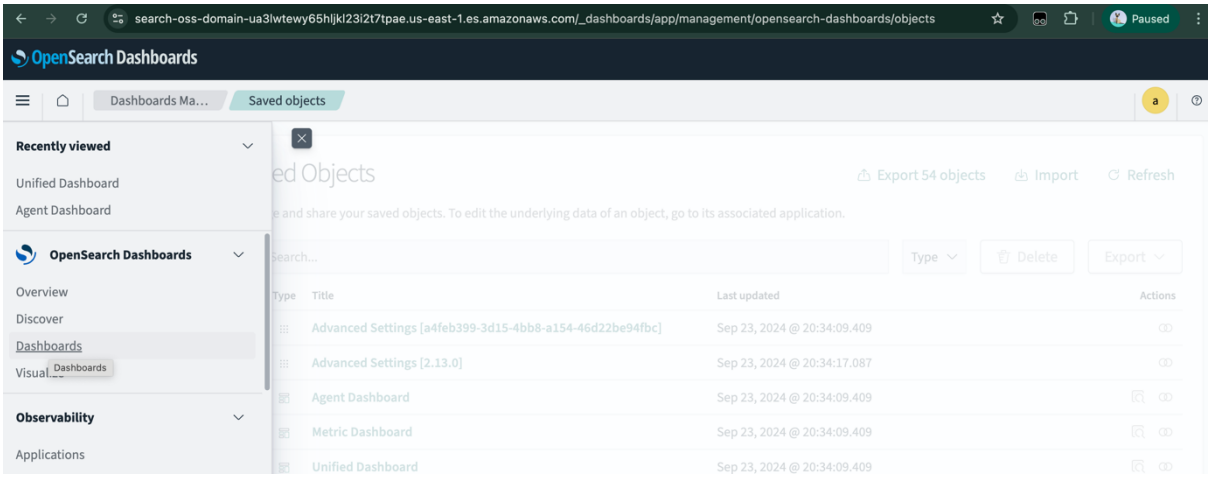
Step 6.1: head to Dashboard Management on the side panel

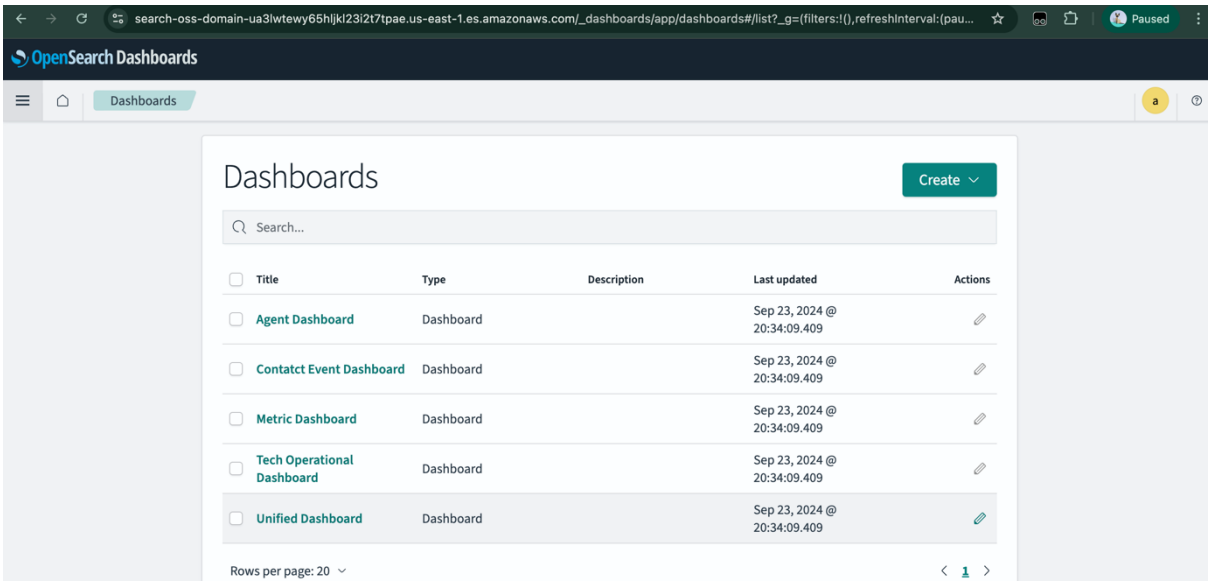


Step 6.2: Import the ndJson file from the below location https://github.com/Fraser27/amazon-connect-analytics/blob/main/connect-rt-aoss-export.ndjson

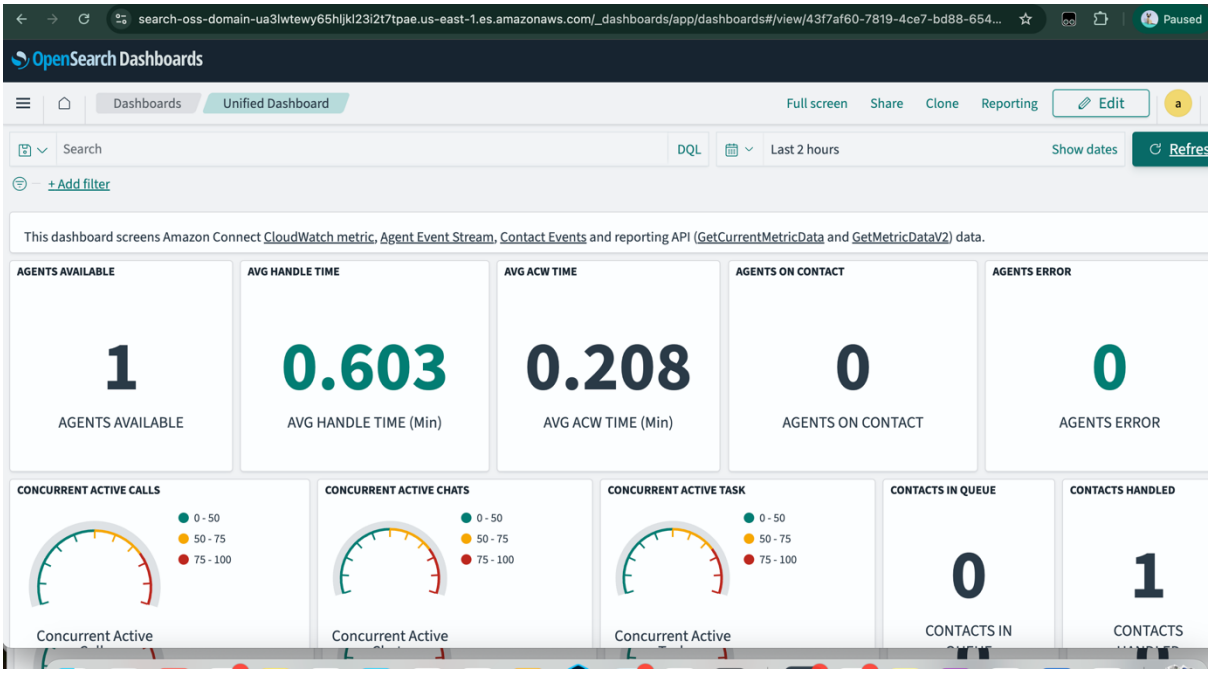


Step 6.3: Head to Dashboard and click on “**Unified Dashboard”.** You should also see multiple dashboards created





Unified Dashboard view



7. Testing: Head to Amazon Connect Agent and start testing. You should see data flowing into Opensearch in real-time. You could now build notification systems on Opensearch to be notified about anomalies in real-time
8. If all works fine, delete the Opensearch serverless collection.

Security

1. How to give Opensearch Dashboard access to users

A. Step 1. Create a resource-based role to be assumed by a select user on your AWS account. The role should have access to Opensearch
Step 2: Give the user permissions to assume the role
Step 3: Map the role on Opensearch dashboard as a backend role, just like we did for Lambda.

2. How to improve further security on Opensearch

A. Reconfigure the Master username/password post the deployment. Do not create any more users on Opensearch. Ensure all access is only through Backend roles

3. How to keep traffic between Opensearch and the Lambda within a VPC

A. Opensearch allows for configuration of VPC endpoints. Once configured, Move your lambdas into the VPC that has the endpoint configured. This ensure's all traffic flows through the private backbone AWS networks