

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/agentix-rag/blob/main/opensearch-cluster.yaml>

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

The screenshot shows the 'Create stack' wizard in the AWS CloudFormation console. The left sidebar shows the navigation menu with 'CloudFormation' selected. The main area shows the 'Create stack' steps: Step 1: Create stack (selected), Step 2: Specify stack details, Step 3: Configure stack options, and Step 4: Review and create. The 'Create stack' section has two tabs: 'Prerequisite – Prepare template' and 'Specify template'. Under 'Prerequisite – Prepare template', there are three options: 'Choose an existing template' (selected), 'Use a sample template', and 'Build from Application Composer'. Under 'Specify template', there are three options: 'Amazon S3 URL', 'Upload a template file' (selected), and 'Sync from Git'. The 'Upload a template file' section shows a file named 'opensearch-cluster.yaml' selected. The 'S3 URL' field at the bottom shows the URL: 'https://s3.us-east-1.amazonaws.com/cf-templates-1orkkb1bs207x-us-east-1/2024-09-24T041255.229Z0ac-opensearch-cluster.yaml'.

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](#)

The screenshot shows the 'Specify stack details' wizard in the AWS CloudFormation console. The left sidebar shows the navigation menu with 'CloudFormation' selected. The main area shows the 'Specify stack details' steps: Step 2: Specify stack details (selected), Step 3: Configure stack options, and Step 4: Review and create. The 'Specify stack details' section has two tabs: 'Provide a stack name' and 'Parameters'. Under 'Provide a stack name', the 'Stack name' field is filled with 'opensearch-connect-analytics'. Under 'Parameters', there are four fields: 'InstanceCount' (filled with '2'), 'InstanceType' (filled with 't3.small.search'), 'OSDomainName' (filled with 'oss-connect-domain'), and 'OSPassword' (filled with 'S!l@19xtewsad'). The 'OSUsername' field is filled with 'admin'. At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next'.

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

CloudFormation < CloudFormation > Stacks > Create stack

Stacks
StackSets
Exports

Application Composer [New](#)
IaC generator

▼ Registry
Public extensions
Activated extensions
Publisher

Spotlight

[Feedback](#)

Step 1: Create stack
Step 2: Specify stack details
Step 3: **Configure stack options**
Step 4: Review and create

Configure stack options

Tags - optional
Tags (key-value pairs) are used to apply metadata to AWS resources, which can help in organising, identifying and categorising those resources. You can add up to 50 unique tags for each stack.
No tags associated with the stack.
[Add new tag](#)
You can add 50 more tag(s)

Permissions - optional
Specify an existing AWS Identity and Access Management (IAM) service role that CloudFormation can assume.

IAM role - optional
Choose the IAM role for CloudFormation to use for all operations performed on the stack.
IAM role name: [Remove](#) [+](#)

⚠ AWS CloudFormation will use this role for all stack operations. Other users that have permissions to operate on this stack will be able to use this role, even if they don't have permission to pass it. Ensure that this role grants the least privilege.

Stack failure options

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

CloudFormation < CloudFormation > Stacks > Create stack

Stacks
StackSets
Exports

Application Composer [New](#)
IaC generator

▼ Registry
Public extensions
Activated extensions
Publisher

Spotlight

[Feedback](#)

Step 1: Create stack
Step 2: Specify stack details
Step 3: Configure stack options
Step 4: **Review and create**

There are no notification options defined

Stack creation options

Timeout
-

Termination protection
Deactivated

Capabilities and transforms

ⓘ **Transforms might require access capabilities**
A transform might add Identity and Access Management (IAM) resources that could provide entities access to make changes to your AWS account. If a transform adds IAM resources, you must acknowledge their capabilities to create or update them. Ensure that you want to create or update the IAM resources, and that they have the minimum required permissions. In addition, if they have customised names, check that the names are unique within your AWS account. [Learn more](#)

☐ I acknowledge that AWS CloudFormation might create IAM resources.

☐ I acknowledge that AWS CloudFormation might create IAM resources with customised names.

☐ I acknowledge that AWS CloudFormation might require the following capability:
CAPABILITY_AUTO_EXPAND

[Create changeset](#) [Cancel](#) [Previous](#) [Submit](#)

Step 1.5: Head to OPENSEARCH and note the OPENSEARCH_ENDPOINT url

The screenshot shows the Amazon OpenSearch Service console. On the left is a navigation menu with sections like 'Managed clusters', 'Serverless', and 'Ingestion'. The main area displays the 'oss-domain' configuration. A red arrow points to the 'Domain endpoint' URL: https://search-oss-domain-ua3lwtewy65Njk123t17tpae-us-east-1.es.amazonaws.com/_dashboards. Other visible information includes the domain name 'oss-domain', domain ARN 'arn:aws:es:us-east-1:300036611303:domain/oss-domain', and the service software version 'OpenSearch_2_13_R20240520-P5 (latest)'.

2. Create an integration pipeline between Amazon Connect and Opensearch

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

The screenshot shows the Amazon S3 console. The bucket name is 'aoss-ac-[redacted]-useast'. The 'Objects' tab is selected, showing a list of objects. The first object is 'source-code.zip', which is a zip file, 17.2 MB in size, and was last modified on September 23, 2024, at 19:51:21 (UTC+05:30). The storage class is 'Standard'.

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

The screenshot shows the Amazon Connect console interface. On the left is a navigation sidebar with options like 'Instances', 'Third-party applications', 'Overview', 'Channels and communications', 'Tasks', 'Telephony', 'Applications', 'Amazon Q', 'Analytics tools', 'Cases', 'Customer Profiles', 'Forecasting, capacity planning, and scheduling', 'Voice ID', 'Approved origins', 'Data storage', 'Data streaming', 'Flows', and 'Documentation'. The main content area is titled 'Account overview' and contains three sections: 'Access information' with an 'Access URL' and a warning box about emergency access; 'Distribution settings' with an 'Instance ARN' (highlighted with a blue box) and a 'Service-linked role'; and 'Tags'.

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

The screenshot shows the AWS CloudFormation console's 'Create stack' wizard. The left sidebar includes 'Stacks', 'StackSets', 'Exports', 'Application Composer', 'laC generator', 'Registry', 'Spotlight', and 'Feedback'. The wizard progress bar shows four steps: 'Create stack' (selected), 'Specify stack details', 'Configure stack options', and 'Review and create'. The main content area is divided into two sections: 'Prerequisite – Prepare template' and 'Specify template'. The 'Prepare template' section has three options: 'Choose an existing template' (selected), 'Use a sample template', and 'Build from Application Composer'. The 'Specify template' section has three options: 'Amazon S3 URL', 'Upload a template file' (selected), and 'Sync from Git'. Below these, there is a text input field for the template file name, which contains 'connect-realtimemeanalytics-opensearch-cft.yaml'. At the bottom, the 'S3 URL' is displayed as 'https://s3.us-east-1.amazonaws.com/cf-templates-1orkkb1bs207x-us-east-1/2024-09-24T033951.29724zt-connect-realtimemeanalytics-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

The screenshot shows the 'Specify stack details' step in the AWS CloudFormation console. The left sidebar contains navigation links for CloudFormation, Stacks, StackSets, Exports, Application Composer, and IaC generator. The main content area is titled 'Specify stack details' and includes a progress bar with four steps: Step 1 (Create stack), Step 2 (Specify stack details), Step 3 (Configure stack options), and Step 4 (Review and create). The 'Specify stack details' section contains three input fields: 'Stack name' (filled with 'amazon-connect'), 'AgentEventDataStreamArn' (empty), and 'SolutionSourceBucket' (filled with 'aoss-ac-...-useast'). A 'Next' button is visible at the bottom right.

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

The screenshot shows the 'Configure stack options' step in the AWS CloudFormation console. The left sidebar is the same as in the previous screenshot. The main content area is titled 'Configure stack options' and includes a progress bar with four steps: Step 1 (Create stack), Step 2 (Specify stack details), Step 3 (Configure stack options), and Step 4 (Review and create). The 'Configure stack options' section contains two input fields: 'IAM role name' (filled with 'cloudform-admin') and 'Permissions - optional' (empty). A 'Next' button is visible at the bottom right.

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

CloudFormation

Stacks

StackSets

Exports

Application Composer [New](#)

laC generator

▼ Registry

Public extensions

Activated extensions

Publisher

Spotlight

Feedback

There are no notification options defined

Stack creation options

Timeout

-

Termination protection

Deactivated

Capabilities and transforms

① Transforms might require access capabilities

A transform might add Identity and Access Management (IAM) resources that could provide entities access to make changes to your AWS account. If a transform adds IAM resources, you must acknowledge their capabilities to create or update them. Ensure that you want to create or update the IAM resources, and that they have the minimum required permissions. In addition, if they have customised names, check that the names are unique within your AWS account. [Learn more](#)

☐ I acknowledge that AWS CloudFormation might create IAM resources.

☐ I acknowledge that AWS CloudFormation might create IAM resources with customised names.

☐ I acknowledge that AWS CloudFormation might require the following capability: CAPABILITY_AUTO_EXPAND

Create changeset

Cancel

Previous

Submit

Step 2.8: Wait for the stack to finish deployment

CloudFormation > Stacks > amazon-conn

Stacks (4)

Filter by stack name

Active

View nested

Stacks

amazon-conn

2024-09-23 20:14:55 UTC+0530

CREATE_COMPLETE

oss-cog-test-2

2024-09-23 18:02:19 UTC+0530

CREATE_COMPLETE

CDKToolkit

2023-11-20 22:31:49 UTC+0530

CREATE_COMPLETE

lambda-trigger

2023-10-15 17:27:45 UTC+0530

CREATE_COMPLETE

amazon-conn

Delete Update Stack actions Create stack

Stack info Events Resources Outputs Parameters Template Chan

Overview

Stack ID

arn:aws:cloudformation:us-east-1:300036611305:stack/amazon-conn/65f66820-79ba-11ef-b9c2-1242c5f8ade9

Description

Amazon OpenSearch Serverless template to create an IAM user, encryption policy, data access policy and collection V1

Status

CREATE_COMPLETE

Detailed status

-

Status reason

-

Root stack

-

Parent stack

-

Created time

2024-09-23 20:14:55 UTC+0530

Updated time

-

Deleted time

-

Drift status

NOT_CHECKED

Last drift check time

-

Termination protection

Deactivated

3. Add the Opensearch Endpoint URL as an environment variable to your lambda functions.

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

Lambda > Functions

Functions (14) Last fetched 14 seconds ago Actions Create function

Q Filter by tags and attributes or search by keyword

<input type="checkbox"/>	Function name	Description	Package type	Runtime	Last modified
<input type="checkbox"/>	amazon-conn-SolutionLambdaSourceFunction	-	Zip	Node.js 18.x	13 hours ago
<input type="checkbox"/>	amazon-conn-BatchLambda	CaseBatchEvent	Zip	Node.js 18.x	13 hours ago
<input type="checkbox"/>	amazon-conn-CloudWatchLambda	CWBatchEvent	Zip	Node.js 18.x	13 hours ago
<input type="checkbox"/>	amazon-conn-AgentLambda	AgentEventLambda	Zip	Node.js 18.x	13 hours ago
<input type="checkbox"/>	amazon-conn-CacheLambda	CacheBatchEvent	Zip	Node.js 18.x	13 hours ago
<input type="checkbox"/>	amazon-conn-ConnectEventLambda	ConnectEventLambda	Zip	Node.js 18.x	13 hours ago

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.

Lambda > Functions > amazon-conn-BatchLambda

amazon-conn-BatchLambda Throttle Copy ARN Actions

► **Function overview** Info Export to Application Composer Download

Code Test Monitor **Configuration** Aliases Versions

General configuration

Triggers

Permissions

Destinations

Function URL

Environment variables

Tags

VPC

RDS databases

Environment variables (5) Edit

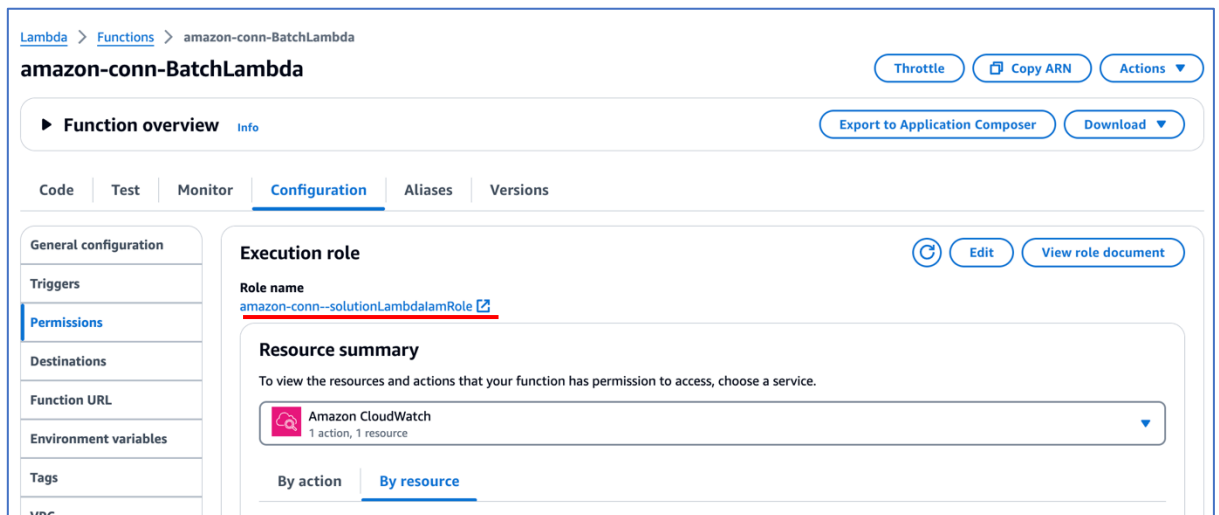
The environment variables below are encrypted at rest with the default Lambda service key.

Q Find environment variables

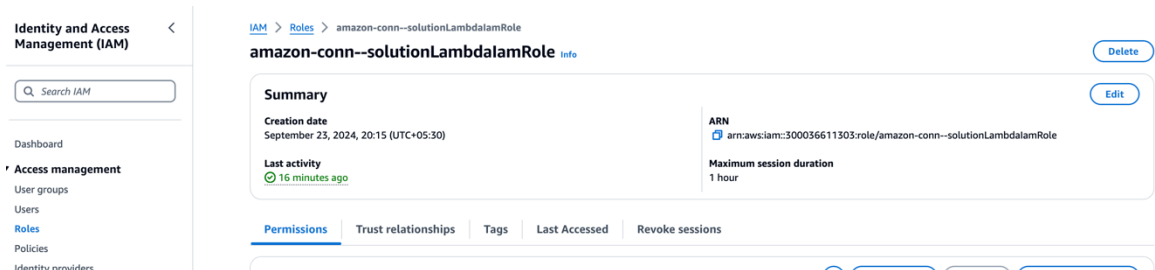
Key	Value
ACCOUNT_ID	300036611303
AOSSDataCacheTable	amazon-conn-AOSSDataCache
CollectionEndpoint	https://zcsxq1dokyvylfk2hmf.us-east-1.aoss.amazonaws.com
ConnectInstanceid	f13ea585-0bbc-4197-a47a-dec2a4ccf6be
<u>OPENSEARCH_ENDPOINT</u>	<u>https://search-oss-domain-ua3lwtey65hjlz3i2t7tpae.us-east-1.es.amazonaws.com</u>

4. Add a Backend role to allow Lambda to access Opensearch

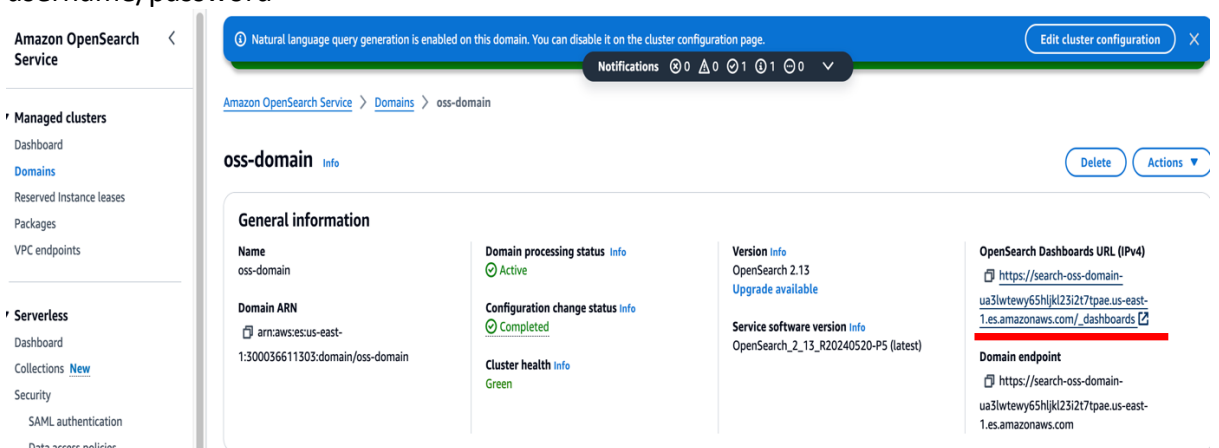
Step 4.1: Identify the LAMBDA role. Click on any lambda function and head to configuration.



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



Log in to OpenSearch Dashboards

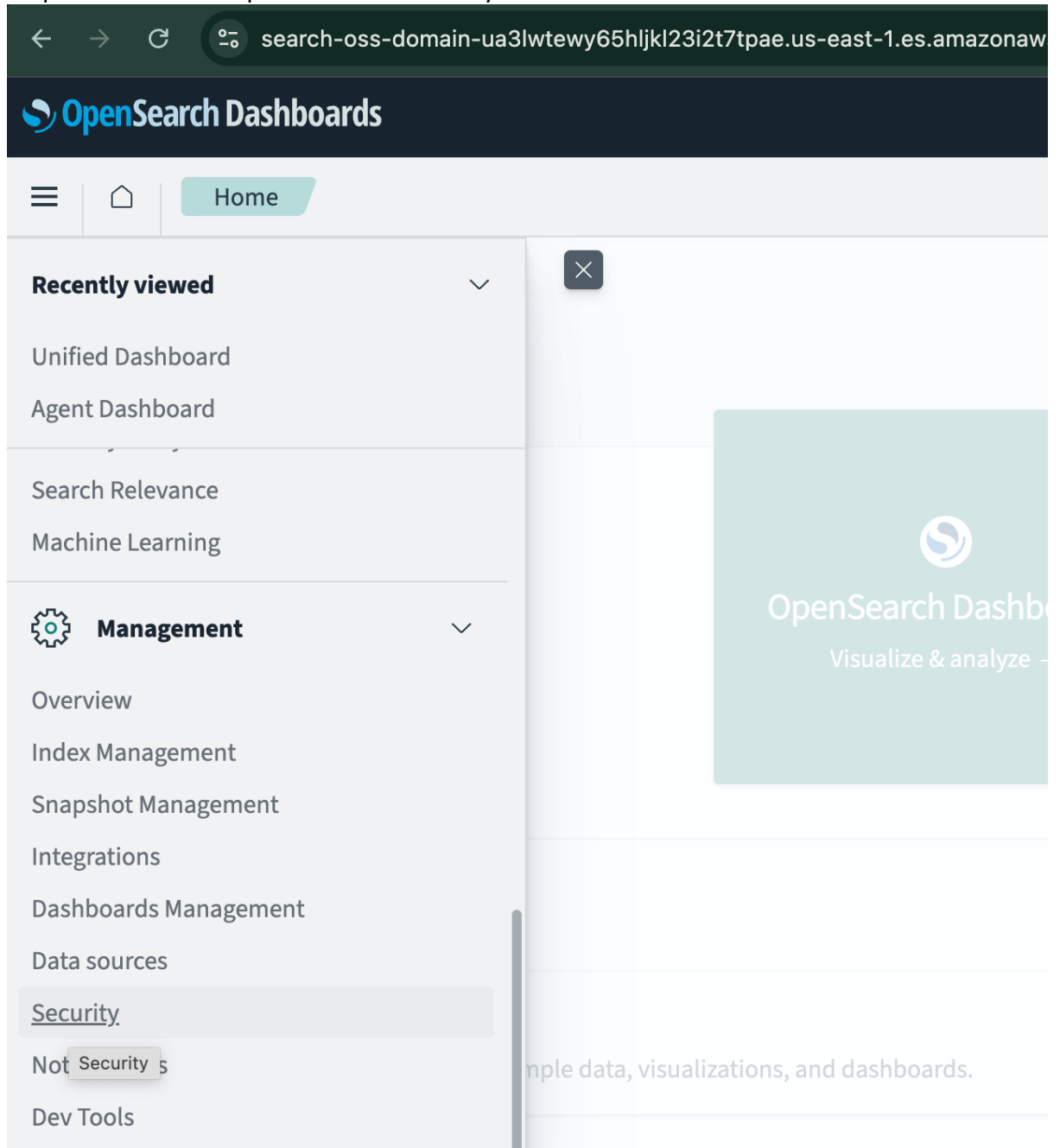
If you have forgotten your username or password, contact your system administrator.

 admin



Log in

Step 4.5: On the side panel click on Security



Step 4.6: Click on Explore existing roles

The screenshot shows the OpenSearch Dashboards Security page. The left sidebar contains a menu with 'Security' selected, and sub-items: 'Get Started', 'Authentication', 'Roles', 'Internal users', 'Permissions', 'Tenants', and 'Audit logs'. The main content area is titled 'Get started' and contains a paragraph: 'The OpenSearch security plugin lets you define the API calls configuration consists of these steps.' Below this is a numbered list starting with '1 Create roles'. A description follows: 'Roles are reusable collections of permissions. The default roles are predefined roles that meet your exact needs. [Learn more](#)'. At the bottom of the main content area are two buttons: 'Explore existing roles' and 'Create new role'.

search-oss-domain-ua3lwtewy65hljkl23i2t7tpae.us-east-1.es.amazonaws.com

OpenSearch Dashboards

Security

Get started

The OpenSearch security plugin lets you define the API calls configuration consists of these steps.

- 1 Create roles**

Roles are reusable collections of permissions. The default roles are predefined roles that meet your exact needs. [Learn more](#)

[Explore existing roles](#) [Create new role](#)

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.

The screenshot shows the OpenSearch Dashboards Roles page. The left sidebar contains a menu with 'Security' selected, and sub-items: 'Get Started', 'Authentication', 'Roles', 'Internal users', 'Permissions', 'Tenants', and 'Audit logs'. The main content area is titled 'Roles' and contains a section 'Roles (1)'. Below this is a search bar with 'all_access' entered. To the right of the search bar are several filter buttons: 'Cluster permissions', 'Index permissions', 'Internal users', 'Backend roles', 'Tenants', and 'Customization'. Below the search bar is a table with the following columns: 'Role', 'Cluster permissions', 'Index permissions', 'Internal users', 'Backend roles', 'Tenants', and 'Customization'. The table contains one row for the 'all_access' role. The 'Internal users' column shows 'admin' and the 'Backend roles' column shows 'arn:aws:iam::300036611303:role/amazon-conn-solutionLambdalaRole'. The 'Customization' column shows 'Reserved'.

search-oss-domain-ua3lwtewy65hljkl23i2t7tpae.us-east-1.es.amazonaws.com/_dashboards/app/security-dashboards-plugin#/roles

OpenSearch Dashboards

Security

Roles

Roles (1)

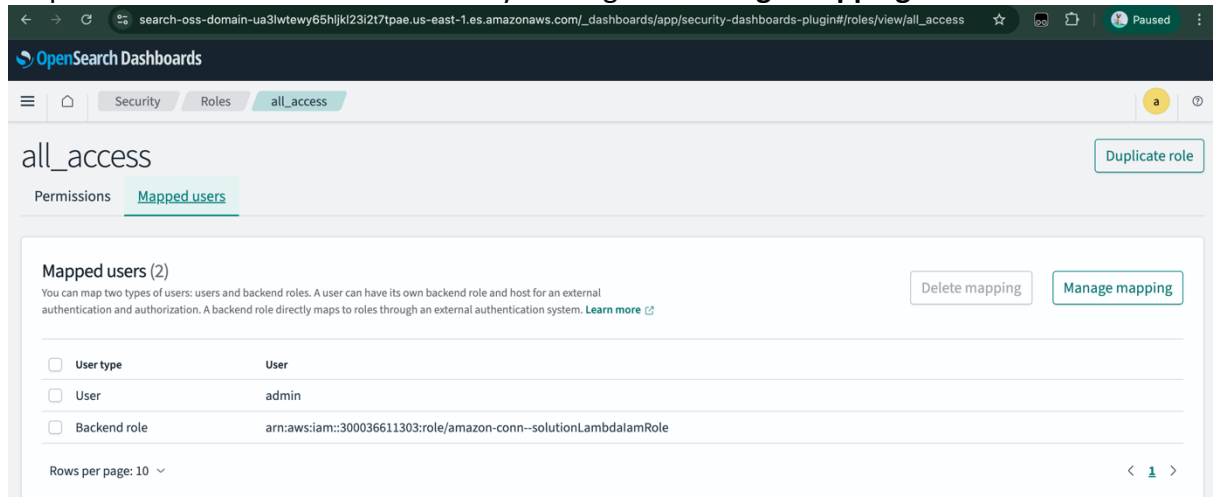
Roles are the core way of controlling access to your cluster. Roles contain any combination of cluster-wide permission, index-specific permissions, document- and field-level security, and tenants. Then you map users to these roles so that users gain those permissions. [Learn more](#)

Search: all_access

Cluster permissions Index permissions Internal users Backend roles Tenants Customization

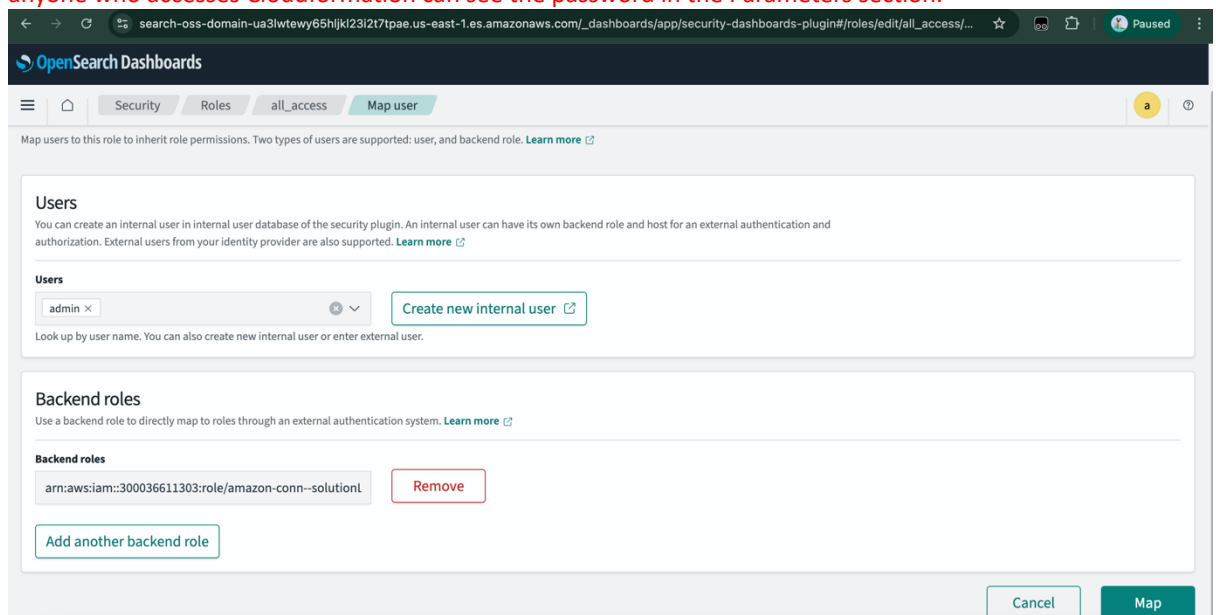
Role	Cluster permissions	Index permissions	Internal users	Backend roles	Tenants	Customization
all_access	*	*	admin	arn:aws:iam::300036611303:role/amazon-conn-solutionLambdalaRole	*	Reserved

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



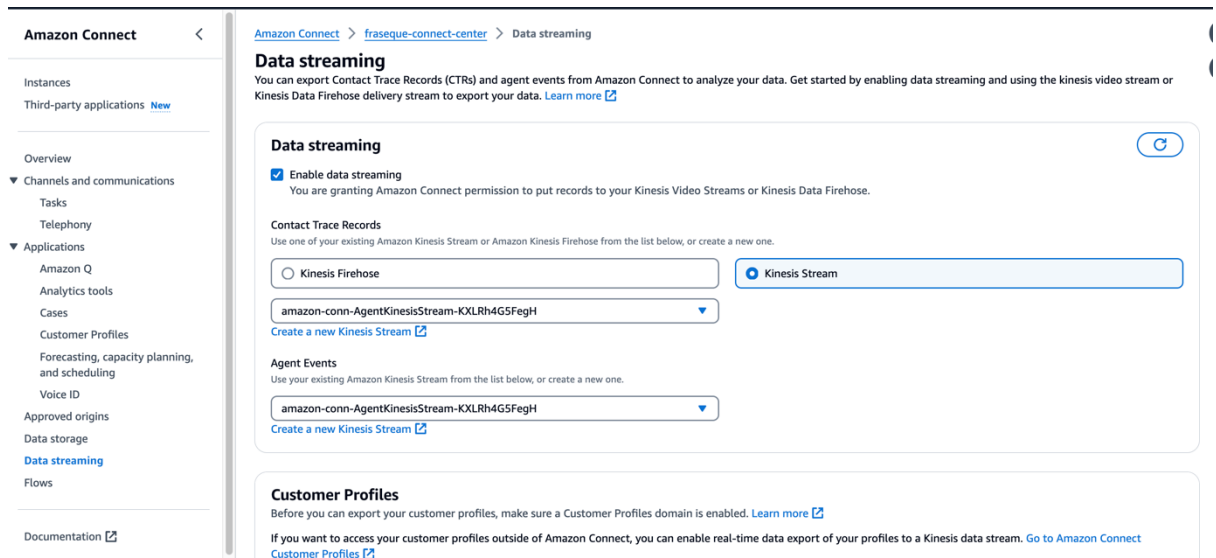
Step 4.9: Manage Mapping page looks as follows

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.



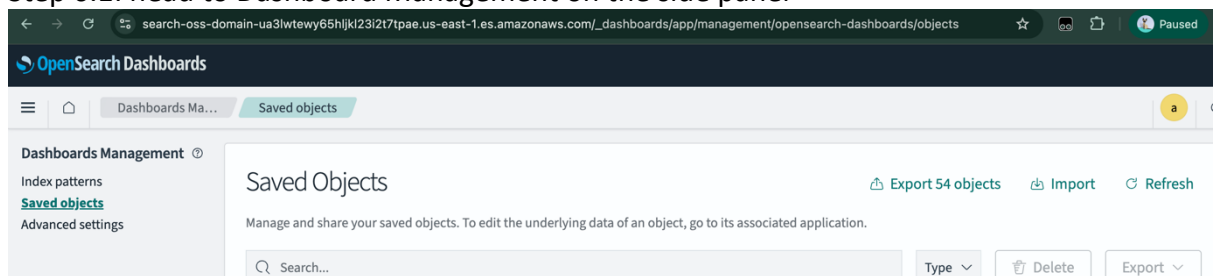
5. Enable Amazon Connect Data Streaming

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.



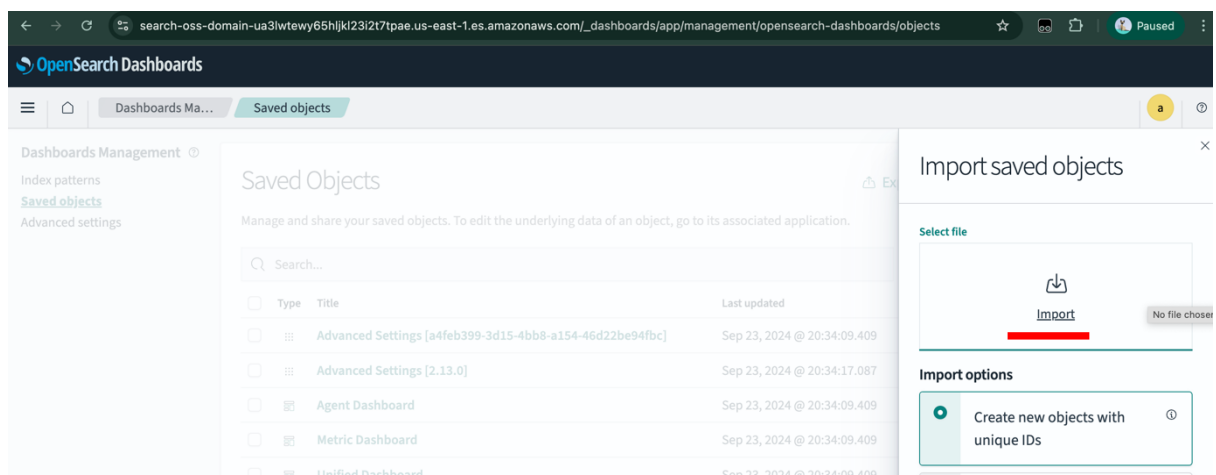
6. Import the dashboard objects into Opensearch as below

Step 6.1: head to Dashboard Management on the side panel

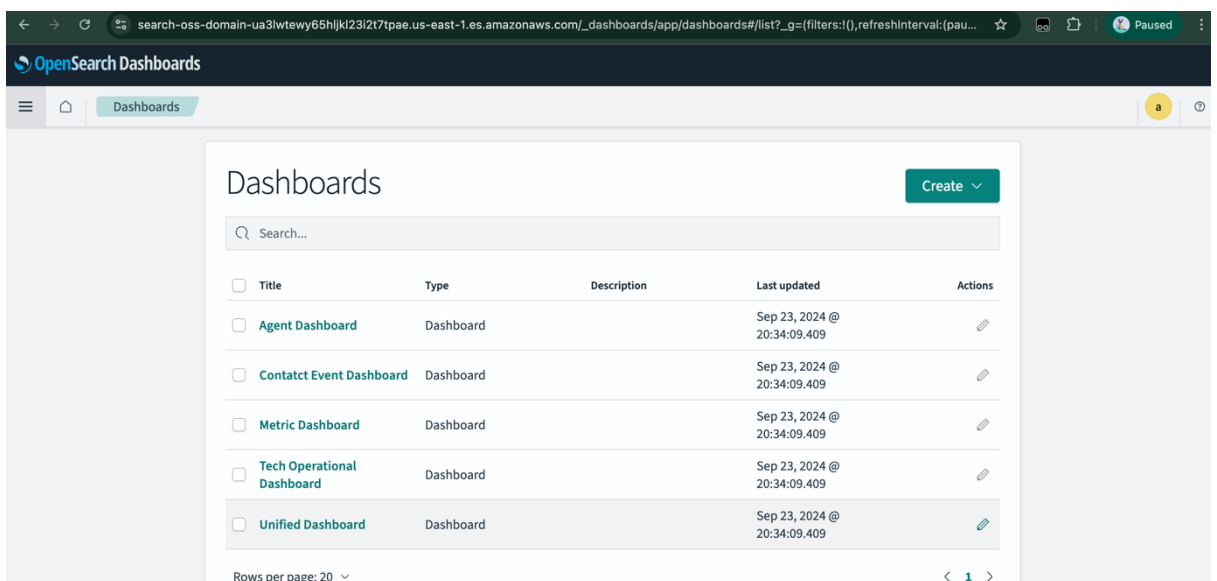
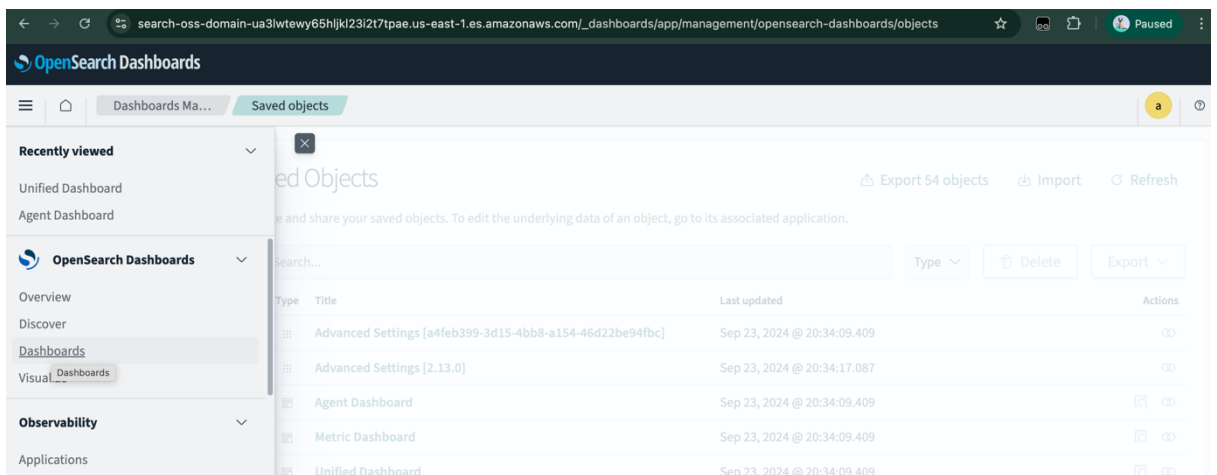


Step 6.2: Import the ndJson file from the below location

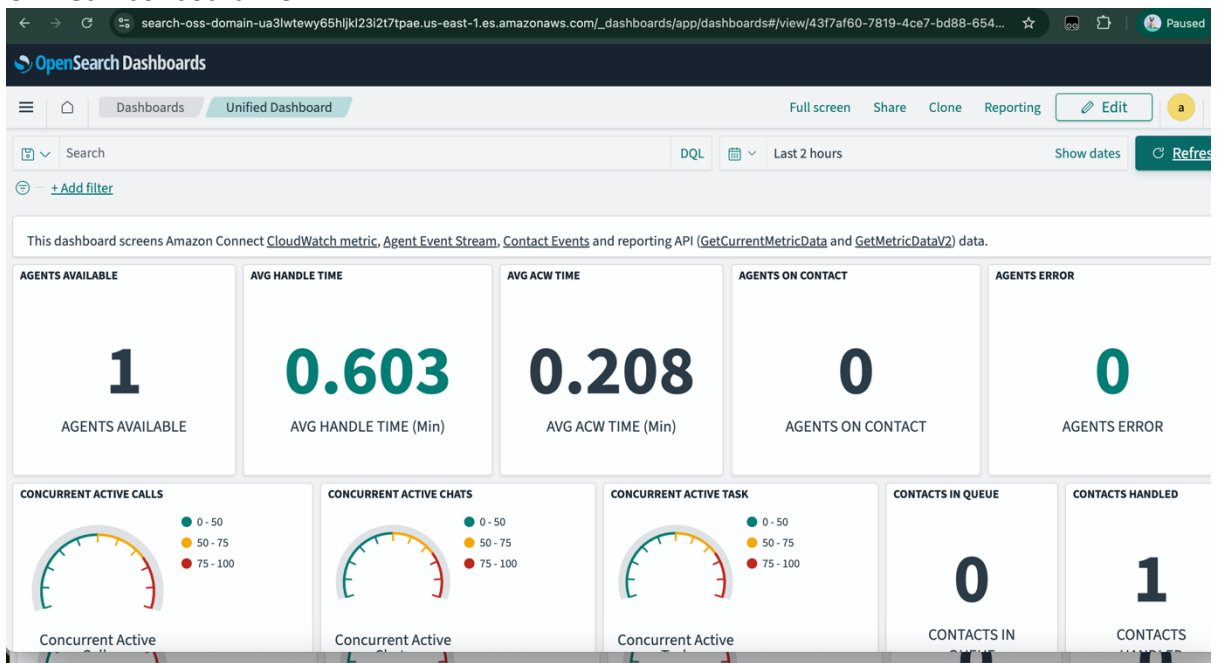
<https://github.com/Fraser27/amazon-connect-analytics/blob/main/connect-rt-aoss-export.ndjson>



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