

Note

The exercises in this course will have an associated charge in your AWS account. In this exercise, you will create the following resources:

- Amazon Kinesis Data Firehose delivery streams
- Amazon Simple Storage Service (Amazon S3) buckets
- Amazon Kinesis data analytics application
- Amazon OpenSearch Service domain
- Amazon Elastic Compute Cloud (Amazon EC2) instance and AWS Identity and Access Management (IAM) role (created by AWS CloudFormation)

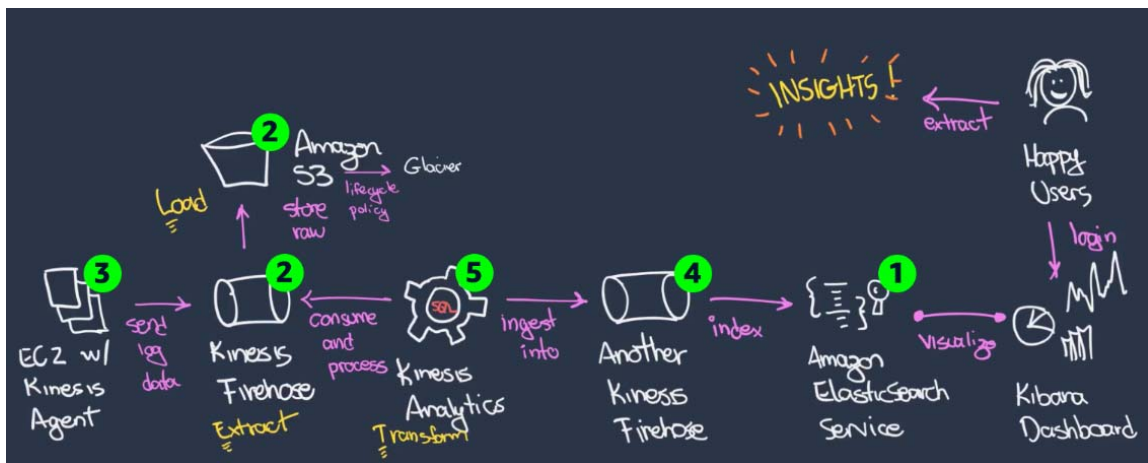
The final exercise task includes instructions to delete all the resources that you create for this exercise.

Familiarize yourself with [Amazon Kinesis Data Streams](#), [Amazon S3 pricing](#), [Amazon Kinesis Data Firehose](#), [Amazon OpenSearch Service](#), and the [AWS Free Tier](#).

Exercise 2: Building a log-analytics solution with Amazon Kinesis

Imagine that you are a systems administrator who is responsible for the health of a large, web-server cluster. As part of your duties, you need to constantly monitor the server access logs for anything unusual. Normally, you would send these logs to a log server and parse the data with your scripting skills. However, the cluster has grown exponentially larger within the last few months. It now requires a more robust solution to keep up with the demand. You are tasked with creating a log-analytics solution on AWS that is extremely scalable.

In this exercise, you produce data with the Kinesis agent, which runs on an EC2 instance. The agent simulates one of the web servers in your organization's large server farm. Then, you ingest some dummy access logs with Kinesis Data Firehose. You move those logs to Amazon S3. Then, you use Kinesis Data Analytics to get data and aggregate data points for Kinesis Data Analytics to output. You send the aggregated data to another Kinesis Data Firehose delivery stream that outputs the data to Amazon OpenSearch Service. Finally, you visualize the data with OpenSearch Dashboards. The following schematic provides an overview of your workflow:



Flowchart of exercise tasks

Setting up

When you sign in to the AWS Management Console, you must first ensure that you have appropriate Identity and Access Management (IAM) users, roles, or policies to work with cloud resources. IAM ensures that only the right users have permissions to perform certain tasks. With IAM, you can securely control access to your account and set up granular permissions on an as-needed basis.

In this exercise, you use the AWS CloudFormation template to configure backend resources. The AWS CloudFormation template is a JSON or YAML file that provisions some of the AWS services for your needs. We provide the template for you later in this section. Before you upload the template in AWS CloudFormation, you must create an IAM role with full administrative privileges for CloudFormation based on the EC2 use case.

1. In the AWS Management Console, enter **IAM** in the search field. Choose **IAM** from the list.
2. In the navigation pane, choose **Roles**.
3. Choose **Create role**.
4. For **Trusted entity type**, choose **AWS service**.
5. For **Use case**, choose **EC2**.
6. Choose **Next**.
7. Under **Permission policies**, in the search field, enter **CloudFormation**.
8. In the list of available options, select the AWS managed policy that provides full access to AWS CloudFormation. Choose **Next**.
9. For **Role name**, enter **CloudFormation**.
10. Choose **Create role**.

With the new IAM role, you now have access to AWS CloudFormation. You can also use IAM roles to share temporary access with users who might need to access the AWS resources associated with your account. For more information about IAM, see [What's IAM](#).

Download the following CloudFormation template: [exercise-2-kinesis.yml](#). Follow the instructions to upload this template in AWS CloudFormation.

Note: If you have an existing virtual private cloud (VPC) with the Classless Inter-Domain Routing (CIDR) block `10.16.0.0/16`, you must edit the template and change its CIDR block.

1. In the search field of the AWS Management Console, enter **CloudFormation**. Choose **CloudFormation** from the list.
2. At the top right of the console, make sure you are in the **US East (N. Virginia) - us-east-1** Region.
3. Choose **Create stack > With new resources (standard)**.
4. Choose **Upload a template file**.
5. Select **Choose file** and browse to where you downloaded the `exercise-2-kinesis` template.
6. Select the file and choose **Open**.
7. Choose **Next**.
8. For **Stack name**, enter `exercise-2-kinesis`.
9. Choose **Next**, and then choose **Next** again.
10. Select the acknowledgement and choose **Create stack**.

Task 1: Creating an Amazon OpenSearch Service cluster

In this task, you create an Amazon OpenSearch Service cluster. OpenSearch Service performs interactive log analytics, real-time application monitoring, and more.

1. In the search field of the AWS Management Console, search for and open **Amazon OpenSearch Service**.
2. Make sure that you are in the **N. Virginia** Region.
3. Choose **Create domain** and configure the following settings:
 - **Domain name:** `web-log-summary`
 - **Deployment type:** *Development and testing*
 - **Version:** The latest version of *OpenSearch*
 - **Data nodes > Instance type:** *m5.large.search*
 - **Network:** *Public access* **Note:** You would typically want to run a production-level workload in a VPC. However, for testing purposes, you use *Public access*. A production cluster can have more restrictive policies, such as restricting IP addresses or hosting the cluster in a private subnet.
 - **Fine-grained access control:** *Enable fine-grained access control*
 - **Master user:** *Create master user*
 - **Master username:** `admin`
 - **Master password** and **Confirm master password:** `#MasterPassword1@`
 - **Access policy:** *Configure domain level access policy*
1. In the **Domain access policy** section, choose the **JSON** tab and paste the following JavaScript Object Notation (JSON) code:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

    "Effect": "Allow",
    "Principal": {
      "AWS": [
        "*"
      ]
    },
    "Action": [
      "es:*"
    ],
    "Resource": "arn:aws:es:us-east-1:<FMI>:domain/web-log-summary/*"
  }
]
}

```

Replace the FMI in the JSON code with your account number. When you replace the FMI with your own value, make sure that you also delete the angle brackets (<>). You can find your account number in the top-right area of the console, next to the Region. It should have a format similar to 0000-0000-0000. Remove the dashes before you save your changes. For example:

```

],
"Resource": "arn:aws:es:us-east-1:000000000000:domain/web-log-summary/*"
}

```

4. Choose **Create**. To create a cluster may take 10 to 15 minutes.

Task 2: Creating the first Amazon Kinesis Data Firehose delivery stream and the Amazon S3 bucket

In this task, you create an Amazon Kinesis Data Firehose delivery stream, which includes creating an Amazon S3 bucket. With Kinesis Data Firehose, you deliver real-time streaming data to Amazon S3.

1. In the search field of the AWS Management Console, search for and open **Kinesis**.
2. Choose **Kinesis Data Firehose**, and choose **Create delivery stream**.
3. Configure the following settings.
 - **Source:** *Direct PUT*
 - **Destination:** *Amazon S3*
 - **Delivery stream name:** web-log-ingestion-stream
 - **Destination settings:** *Create*

Choosing **Create** for **Destination settings** opens a separate browser tab so that you can create an S3 bucket.

4. For **Bucket name**, enter a *unique* name.

As an example, you can use the following naming convention and replace the FMI with your initials.

```
<FMI>-web-log-ingestion-bucket
```

Example:

```
emr-web-log-ingestion-bucket
```

5. For **Region**, keep **US East (N. Virginia) us-east-1**. Your OpenSearch Service cluster and Kinesis streams are provisioned in this Region.
6. Choose **Create bucket**.
7. Switch back to the **Kinesis Data Firehose** console.
8. Next to the **S3 bucket** box, choose **Browse** and select the S3 bucket that you created. If the bucket isn't listed, choose the refresh icon.
9. After you select the bucket, click **Choose**.
10. Choose **Create delivery stream**. *This process can take up to 5 minutes to complete.*

Task 3: Installing the Kinesis Agent

In this task, you install the Kinesis Agent on the EC2 instance. Kinesis Agent is a script that runs in the background to generate server access logs. You can look at the access logs in the `/tmp/logs` directory.

1. In the search field of the AWS Management Console, search for and open **EC2**.
2. In the **Resources** pane, choose **Instances (running)**.

A Dummy Web Server instance should already be running.

3. Open the instance summary information of the *Dummy Web Server* instance by choosing its **Instance ID**.

4. Choose **Connect**.

5. In **Connect to instance**, choose the **Session Manager** tab and choose **Connect**.

This action loads a session, and you should be presented with a shell prompt: `sh-4.2$`.

6. In the session terminal, install the Kinesis Agent on the instance by running the following command:

```
sudo yum install -y aws-kinesis-agent
```

It may take a minute for the command to start installing the Kinesis Agent.

1. Confirm that you want to install the required packages by pressing **Y** on your keyboard.

Next, you will edit the `agent.json` file by replacing the value of the **deliveryStream** key with name of the delivery stream that you created.

2. In your text editor of choice, open the `/etc/aws-kinesis/agent.json` file. The following example uses Vim:

```
sudo vim /etc/aws-kinesis/agent.json
```

3. View the current contents of `agent.json`, which should be similar to this example:

```
{
  "cloudwatch.emitMetrics": true,
  "kinesis.endpoint": "",
  "firehose.endpoint": "",

  "flows": [
    {
      "filePattern": "/tmp/app.log*",
      "kinesisStream": "yourkinesisstream",
      "partitionKeyOption": "RANDOM"
    },
    {
      "filePattern": "/tmp/app.log*",
      "deliveryStream": "yourdeliverystream"
    }
  ]
}
```

4. Use the keyboard arrows to navigate the file and enter the `i` key to start editing. Delete existing contents of `agent.json` and paste the following code. Make sure that you update the FMI value for **deliveryStream** (which should be `web-log-ingestion-stream`).

```
{
  "cloudwatch.emitMetrics": true,
  "kinesis.endpoint": "",
  "firehose.endpoint": "",
  "flows": [{
    "filePattern": "/tmp/logs/access_log*",
    "deliveryStream": "<FMI>",
    "dataProcessingOptions": [{
      "optionName": "LOGTOJSON",
      "logFormat": "COMMONAPACHELOG"
    }]
  }]
}
```

Note: If you gave the delivery stream a different name, exit the `agent.json` file and retrieve the name of your delivery stream by running the following command:

```
aws firehose list-delivery-streams --region us-east-1
```

```
{
  "DeliveryStreamNames": [
    "web-log-ingestion-stream"
  ],
  "HasMoreDeliveryStreams": false
}
```

The updated `agent.json` file should have the following:

```
{
  "cloudwatch.emitMetrics": true,
  "kinesis.endpoint": "",
  "firehose.endpoint": "",
  "flows": [{
    "filePattern": "/tmp/logs/access_log*",
    "deliveryStream": "web-log-ingestion-stream",
    "dataProcessingOptions": [{
      "optionName": "LOGTOJSON",
      "logFormat": "COMMONAPACHELOG"
    }]
  }]
}
```

5. Enter the **ESC** key to switch back to command mode. If you are using Vim, enter `:wq` to save and exit the file.
6. Start the agent by running the following command:

```
sudo service aws-kinesis-agent start
```

If you want to see what is happening, or if you want to troubleshoot the agent, you can use the `tail -f` command on the `/var/log/aws-kinesis-agent/aws-kinesis-agent.log` file.

Note: It might take approximately 5 minutes before data starts to appear in your Firehose delivery stream.

Task 4: Creating the second Kinesis Data Firehose delivery stream and configuring the OpenSearch Service domain

In this task, you create the second Kinesis Data Firehose delivery stream and another Amazon S3 bucket. This delivery stream sends the data to the OpenSearch Service domain that you created previously.

1. Return to the **Amazon Kinesis** console.
2. On the **Data Firehose** card, choose **Create delivery stream** and configure the following settings.
 - **Source:** Direct PUT
 - **Destination:** Amazon OpenSearch Service
 - **Delivery stream name:** `web-log-aggregated-data`
 - **Destination settings:** Browse to and select the `web-log-summary` domain
 - **Index:** `request_data`
 - **Backup settings:** *Create*

Again, this action opens a new browser tab so that you can create an Amazon S3 bucket.

3. For **Bucket name**, enter another unique name.

For example, you can create a bucket name by using the following naming convention and replacing the FMI with your initials.

```
<FMI>-web-log-aggregated-errors
```

Example:

```
emr-web-log-aggregated-errors
```

4. For **Region**, keep **US East (N. Virginia) us-east-1**.
5. Choose **Create bucket**.
6. Switch back to the **Kinesis Data Firehose** console.
7. Next to the **S3 backup bucket** box, choose **Browse**.
8. Select the bucket that you created for this stream, and then **Choose**.

Note: If the bucket isn't listed, choose the refresh icon.

9. Expand **Buffer hints, compression and encryption**, and configure the following settings.
 - **Buffer size:** Change to **1**
 - **Buffer interval:** Keep **60**

The frequency of data delivery to Amazon S3 is determined by the Amazon S3 **Buffer size** and **Buffer interval** values that you configure for your delivery stream. Amazon Kinesis provides the fastest data delivery if you configure the lowest buffer size.

10. Choose **Create delivery stream** and wait for the **Status** to become *Active*.
11. In the navigation pane of the AWS Management Console, choose **Services**, and search for and open **IAM**.
12. In the IAM navigation pane, under **Access management**, choose **Roles**.
13. In the search box, enter `KinesisFirehoseServiceRole-web-log-aggre` and choose the **Role name** link.
14. Copy the role's **ARN** value, which should look similar to the following:

```
arn:aws:iam::000000000000:role/service-role/KinesisFirehoseServiceRole-web-log-aggre
```

15. Return to the **OpenSearch Service** console, and in the navigation pane, choose **Domains**.
16. In the **Domains** pane, choose the **web-log-summary** name link.
17. Choose the link for the **OpenSearch Dashboards URL**. This action launches the dashboard.
18. Log in to the dashboard with your OpenSearch Service information:
 - **username:** `admin`
 - **password:** `#MasterPassword1@`

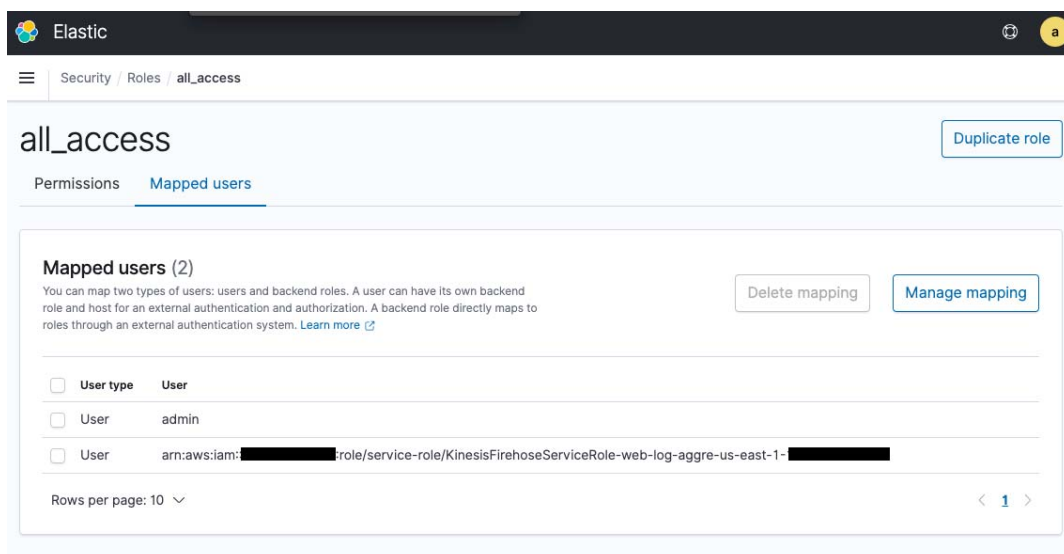
19. If you see a **Welcome to Elastic** dialog box, close the message by choosing **Explore on my own**.
20. If you see a **Select your tenant** dialog box, keep the **Private** setting, and choose **Confirm**.
21. Expand the navigation pane by choosing the menu icon (in the upper-left area of the OpenSearch Dashboards console).
22. In the navigation pane, under **OpenSearch Plugins**, choose **Security**.
23. In the **Security** navigation pane, choose **Explore existing roles**.
24. Search for the *all_access* role and open its details by choosing the **Role** link.
25. Choose the **Mapped users** tab and choose **Manage mapping**.
26. In the **Users box**, paste the *IAM role ARN* that you copied earlier and press **Enter**.

It should look similar to the following example:

```
arn:aws:iam::000000000000:role/service-role/KinesisFirehoseServiceRole-web-log-aggre
```

27. Choose **Map**.

The **Mapped users** list should have an *admin* user and a user with the *IAM role ARN* that you copied. They should look similar to this example:



User mapping list

Don't close the OpenSearch Dashboards. You'll return to it later.

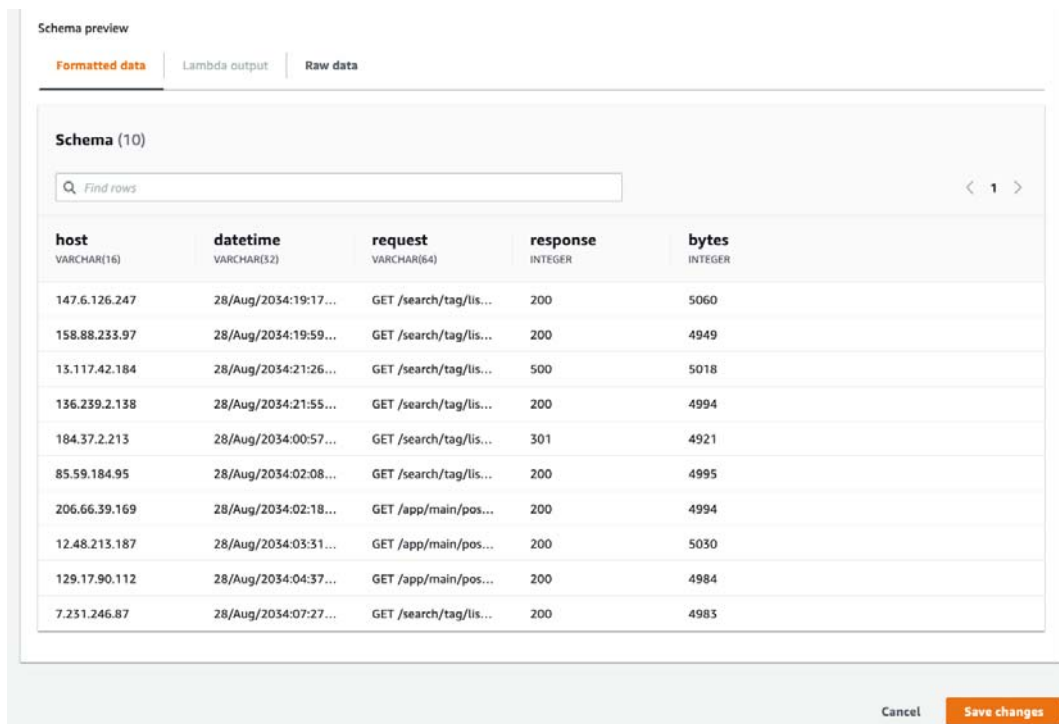
Task 5: Creating the Kinesis Data Analytics application

In this task, you process and analyze streaming data using standard SQL.

1. In a new browser tab, switch back to the **AWS Management Console**.
2. In the console, return to the **Amazon Kinesis** dashboard.
3. In the navigation pane, choose **Analytics applications**.

4. Again in the navigation pane, choose **SQL applications (legacy)**.
5. Choose the **Create SQL application (legacy)** button.
6. For **Application name**, paste `web-log-aggregation-app` and choose **Create legacy SQL application**.
7. Scroll to the **Source** tab, and in the **Source stream** section, and choose **Configure**.
8. Configure the following settings.
 - **Source:** *Kinesis Data Firehose delivery stream*
 - **Delivery stream:** *Browse and select web-log-ingestion-stream (which is the delivery stream you created in Task 2)*
 - **Schema:** *Discover schema and wait for the schema to populate*

You should see a schema that's similar to this example:



Schema preview

Formatted data | Lambda output | Raw data

Schema (10)

Find rows

host	datetime	request	response	bytes
VARCHAR(16)	VARCHAR(32)	VARCHAR(64)	INTEGER	INTEGER
147.6.126.247	28/Aug/2034:19:17...	GET /search/tag/lis...	200	5060
158.88.233.97	28/Aug/2034:19:59...	GET /search/tag/lis...	200	4949
13.117.42.184	28/Aug/2034:21:26...	GET /search/tag/lis...	500	5018
136.239.2.138	28/Aug/2034:21:55...	GET /search/tag/lis...	200	4994
184.37.2.213	28/Aug/2034:00:57...	GET /search/tag/lis...	301	4921
85.59.184.95	28/Aug/2034:02:08...	GET /search/tag/lis...	200	4995
206.66.39.169	28/Aug/2034:02:18...	GET /app/main/pos...	200	4994
12.48.213.187	28/Aug/2034:03:31...	GET /app/main/pos...	200	5030
129.17.90.112	28/Aug/2034:04:37...	GET /app/main/pos...	200	4984
7.231.246.87	28/Aug/2034:07:27...	GET /search/tag/lis...	200	4983

Cancel Save changes

Example schema

9. Choose **Save changes**.
10. At the top of the **web-log-aggregation-app** pane, expand **Steps to configure your application**.
11. In **Step 2**, choose **Configure SQL**. With the SQL editor, you can write queries to process streaming data.
12. In the **SQL code** box, paste the following:

```
CREATE OR REPLACE STREAM "DESTINATION_SQL_STREAM"
(datetime TIMESTAMP, status INTEGER, statusCount INTEGER);

CREATE OR REPLACE PUMP "STREAM_PUMP" AS INSERT INTO "DESTINATION_SQL_STREAM"
SELECT STREAM ROWTIME as datetime, "response" as status, COUNT(*) AS statusCount
FROM "SOURCE_SQL_STREAM_001"
```

```
GROUP BY "response",  
FLOOR(("SOURCE_SQL_STREAM_001".ROWTIME - TIMESTAMP '1970-01-01 00:00:00') minute / 1
```

13. Choose **Save and run application**.

The application should start after 30–90 seconds. You should see the message *Application web-log-aggregation-app has been successfully started*.

You have now configured the application source and real-time analytics.

14. Choose the **web-log-aggregation-app** page.

15. In the main pane, choose the **Destinations** tab.

16. Choose **Add destinations** and configure the following settings.
 - **Destination:** *Kinesis Data Firehose delivery stream*
 - **Delivery stream:** *Browse to and select web-log-aggregated-data*
 - **Connect in-application stream:** *Keep Choose an existing in-application stream selected*
 - **In-application stream name:** *DESTINATION_SQL_STREAM*

17. Choose **Save changes**.

Task 6: Visualizing the data in OpenSearch Dashboards

1. Return to the **OpenSearch Dashboards**.
2. Expand the navigation pane by choosing the menu icon.
3. In the navigation pane, choose **Dashboard**.
4. Choose **Install some sample data**.
5. On the **Sample web logs** tile, choose **Add data**. For more information on how to explore data in OpenSearch, see [Getting started with OpenSearch Dashboards](#).

This solution uses Amazon Kinesis to ingest the raw data, and then saves the data in Amazon S3. It delivers only a refined version of the data to OpenSearch Service.

Depending on how you want to architect your data lake solution—and taking many other subjects into consideration, such as budget included—you might want to take a different approach. For example, you might send the raw logs directly to OpenSearch Service, and then do all the filtering in OpenSearch Dashboards. You can decide how you will design your solution, and remember that it's important to match the workload to the need.

Cleaning up

In this task, you delete the AWS resources that you created for this exercise.

1. Delete the CloudFormation stack.
 - Open the **AWS CloudFormation** dashboard.
 - Delete the **exercise-2-kinesis** stack, and confirm the deletion.
2. Delete the OpenSearch Service domain.
 - Open the **Amazon OpenSearch** dashboard.
 - Choose **web-log-summary**, choose **Delete**, and confirm the deletion.
3. Delete the Kinesis resources.

- Open the **Kinesis** dashboard.
 - In the navigation pane, choose **Delivery streams**.
 - Delete the following delivery streams, and confirm their deletion:
 - **web-log-aggregated-data**
 - **web-log-ingestion-stream**
 - In the navigation pane, choose **Analytics applications** and then choose **SQL applications (legacy)**.
 - Delete **web-log-aggregation-app**, and confirm the deletion.
4. Delete the S3 buckets.
- Open the **Amazon S3** dashboard.
 - Empty and delete the following buckets, and confirm their deletion:
 - **-web-log-aggregated-errors**
 - **-web-log-ingestion-bucket**
5. Delete the IAM roles.
- Open the **IAM** dashboard.
 - In the navigation pane, choose **Roles**.
 - Delete the **Kinesis** roles, and confirm their deletion:
 - **kinesis-analytics-web-log-aggregation-app-us-east-1**
 - **KinesisFirehoseServiceRole-web-log-aggre-us-east-1-xxxxxxxxxxxxxx**
 - **KinesisFirehoseServiceRole-web-log-inges-us-east-1-xxxxxxxxxxxxxx**

© 2022 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections, feedback, or other questions? Contact us at <https://support.aws.amazon.com/#/contacts/aws-training>. All trademarks are the property of their owners.