MINI PROJECT

A REPORT ON

Build a log Analytics Solution on AWS

Team Member:

Adesh Kumar(171500015)

Ayush Kumar Singh(17150075)

Ishan Rathi(171500144)

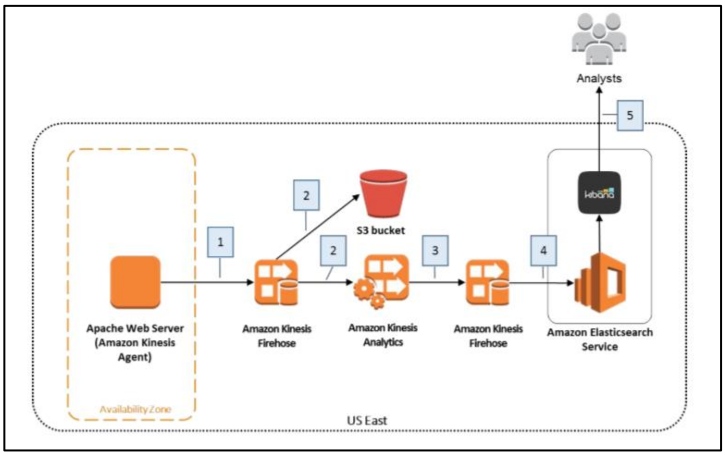
Project Guide:

Dr. Manoj Varshney

Index

ABSTRACT

One of the major benefits to using Amazon Kinesis Analytics is that an entire analysis infrastructure can be created with a serverless architecture. The system created in this tutorial will implement Amazon Kinesis Firehose, Amazon Kinesis Analytics, and Amazon Elasticsearch Service (Amazon ES). Each of these services is designed for seamless integration with one another. The architecture is depicted below.



The web server in this example will be an Amazon Elastic Compute Cloud (EC2) instance. You will install the Amazon Kinesis Agent on this Linux instance, and the agent will continuously forward log records to an Amazon Kinesis Firehose delivery stream (step 1). Amazon Kinesis Firehose will write each log record to Amazon Simple Storage Service (Amazon S3) for durable storage of the raw log data (step 2), and the Amazon Kinesis Analytics application will continuously run an SQL statement against the streaming input data (step 2). The Amazon Kinesis Analytics application will create an aggregated data set every minute and output that data to a second Firehose delivery stream (step 3). This Firehose delivery stream will write the aggregated data to an Amazon ES domain (step 4). Finally, you will create a view of the streaming data using Kibana to visualize the output of your system (step 5).

1. Introduction

1.1 Project Description

Amazon Kinesis Analytics is the easiest way to process streaming data in real time with standard SQL without having to learn new programming languages or processing frameworks. Amazon Kinesis Analytics enables you to create and run SQL queries on streaming data so that you can gain actionable insights and respond to your business and customer needs promptly.

This implementation walks you through the process of ingesting streaming log data, aggregating that data, and persisting the aggregated data so that it can be analyzed and visualized. You will create a complete end-to-end system that integrates several AWS services. You will analyze a live stream of Apache access log data and aggregate the total request for each HTTP response type every minute. To visualize this data in near real-time, you will use a user interface (UI) tool that will chart the results.

1.2 Functional Specifications

Amazon EC2*:* Amazon EC2 provides the virtual application servers, known as instances, to run your web application on the platform you choose. EC2 allows you to configure and scale your compute capacity easily to meet changing requirements and demand. It is integrated into Amazon's computing environment, allowing you to leverage the AWS suite of services.

Amazon Kinesis Firehose*:* Amazon Kinesis Firehose is a fully managed service for delivering real-time streaming data to destinations such as Amazon S3, Amazon Redshift, or Amazon ES. With Firehose, you do not need to write any applications or manage any resources. You configure your data producers to send data to Firehose and it automatically delivers the data to the destination that you specified.

Amazon S3*:* Amazon S3 provides secure, durable, and highly-scalable cloud storage for the objects that make up your application. Examples of objects you can store include source code, logs, images, videos, and other artifacts that are created when you deploy your application. Amazon S3 makes it is easy to use object storage with a simple web interface to store and retrieve your files from anywhere on the web, meaning that your website will be reliably available to your visitors.

Amazon Kinesis Analytics*:* Amazon Kinesis Analytics is the easiest way to process and analyse streaming data in real-time with ANSI standard SQL. It enables you to read data from Amazon Kinesis Streams and Amazon Kinesis Firehose, and build stream processing queries that filter, transform, and aggregate the data as it arrives.

Amazon Elasticsearch Service*:* Amazon ES is a popular open-source search and analytics engine for big data use cases such as log and click stream analysis. Amazon ES manages the capacity, scaling, patching, and administration of Elasticsearch clusters for you while giving you direct access to the Elasticsearch API.

1.3 HARDWARE REQUIREMENT

1.3.1 Processor: i3 Processor and above

1.3.2 Operating System: Windows

1.3.3 RAM: 4 GB and above

1.3.4 Hard Disk: 500 GB and above

1.4 SOFTWARE SPECIFICATION

1.4.1 Technology Implemented: AWS, Cloud Computing, Virtualization

1.4.2 Language Used: Sql

1.4.3 Web Browser: Any Browser

2. Problem Definition

Without log analysis tools, not only would you have to identify which server the log you need is on, but then you'd have to identify which log file it might be in. Once you figured that out, you would then be able to concentrate on getting into the file and searching for clues. However, simply opening the file can be a challenge in itself. A text file that is gigabytes in size can be considered *big data* for most desktops. Though the logs contain what you need, locating the server, identifying the log and searching for the data is simply too much to take on.

3. Objective

We want to track user activities on the networks, sensors, websites, etc for fraud detection, false surfing of users on above platforms, interruptions in between networks, etc. We can also take this project for E-commerce companies for customer analysis.

4. Implementation Details

You can evaluate the simplicity and effectiveness of Amazon Kinesis Analytics with this tutorial, which will walk you through a very simple Amazon Kinesis Analytics application.

You will perform the following steps:

- Step 1: Set Up Prerequisites
- Step 2: Create an Amazon Kinesis Firehose Delivery Stream
- Step 3: Install and Configure the Amazon Kinesis Agent
- Step 4: Create an Amazon Elasticsearch Service Domain
- Step 5: Create a Second Amazon Kinesis Firehose Delivery Stream
- Step 6: Create an Amazon Kinesis Analytics Application
- Step 7: View the Aggregated Streaming Data
- Step 8: Clean Up

This tutorial is not meant for production environments and does not discuss options in depth. After you complete the steps, you can find more in-depth information to create your own Amazon Kinesis Analytics application in the Additional Resources section.

Step 1: Set Up Prerequisites

Before you begin analyzing your Apache access logs with Amazon Kinesis Analytics, make sure you complete these prerequisites:

- Create an AWS Account
- Start an EC2 Instance
- Prepare Your Log Files

This tutorial assumes that the AWS resources have been created in the US East AWS region (us-east-1).

Create an AWS Account

If you already have an AWS account, you can skip this prerequisite and use your existing account. To create AWS account if you do not already one:

1. Go to http://aws.amazon.com/.
2. Choose **Create an AWS Account**.
3. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Start an EC2 Instance

The steps outlined in this implementation assume that you are using an EC2 instance as the web server and log producer. You can use an existing EC2 instance launched from Amazon Linux AMI (version 2015.09 or later) or Red Hat Enterprise Linux (version 7 or later). Otherwise, you can follow the guide Getting Started with Amazon EC2 Linux Instances to start a new instance.[7] If you create a new instance, choose **Amazon Linux AMI** for your operating system. An instance type of **t2.micro** is sufficient for this tutorial.

You also want to ensure that your EC2 instance has an AWS Identity and Access Management (IAM) role configured with permission to write to Amazon Kinesis Firehose and Amazon CloudWatch. For more information, see IAM Roles for Amazon EC2.[8]

Once you have launched the EC2 instance, you will need to connect to it via SSH. To connect to your instance, follow the guide Connect to Your Linux Instance.[9]

Prepare Your Log Files

Because Amazon Kinesis Analytics can analyze your streaming data in near realtime, this tutorial is much more effective when a live stream of Apache access log data is used. If your EC2 instance is not serving HTTP traffic, you will need to generate continuous sample log files.

To create a continuous stream of log file data on your EC2 instance, download, install, and run the Fake Apache Log Generator from Github.[10] Follow the instructions on the project page and configure the script for infinite log file generation.

> Take note of the path to the log file. You will need to know the log file name and path later in this tutorial.

Step 2: Create an Amazon Kinesis Firehose Delivery Stream

In Step 1, you created log files on your web server. Before they can be analyzed with Amazon Kinesis Analytics (Step 6), the log data must first be loaded into AWS. Amazon Kinesis Firehose is a fully managed service for delivering real-time streaming data to destinations such as Amazon Simple Storage Service (Amazon S3), Amazon Redshift, or Amazon Elasticsearch Service (Amazon ES).

In this step, you will create an Amazon Kinesis Firehose delivery stream to save each log entry in Amazon S3 and to provide the log data to the Amazon Kinesis Analytics application that you will create later in this tutorial.

To create the Amazon Kinesis Firehose delivery stream:

1. Open the Amazon Kinesis console at https://console.aws.amazon.com/kinesis.
2. Click **Go to Firehose**.
3. Click **Create Delivery Stream**.
4. On the **Destination** screen:
   a. For Destination, choose Amazon S3.

   b. For **Delivery stream name**, enter web-log-ingestion-stream.

   c. For **S3 bucket**, choose **Create New S3 Bucket**.

      You will be prompted to provide additional information for the new bucket:

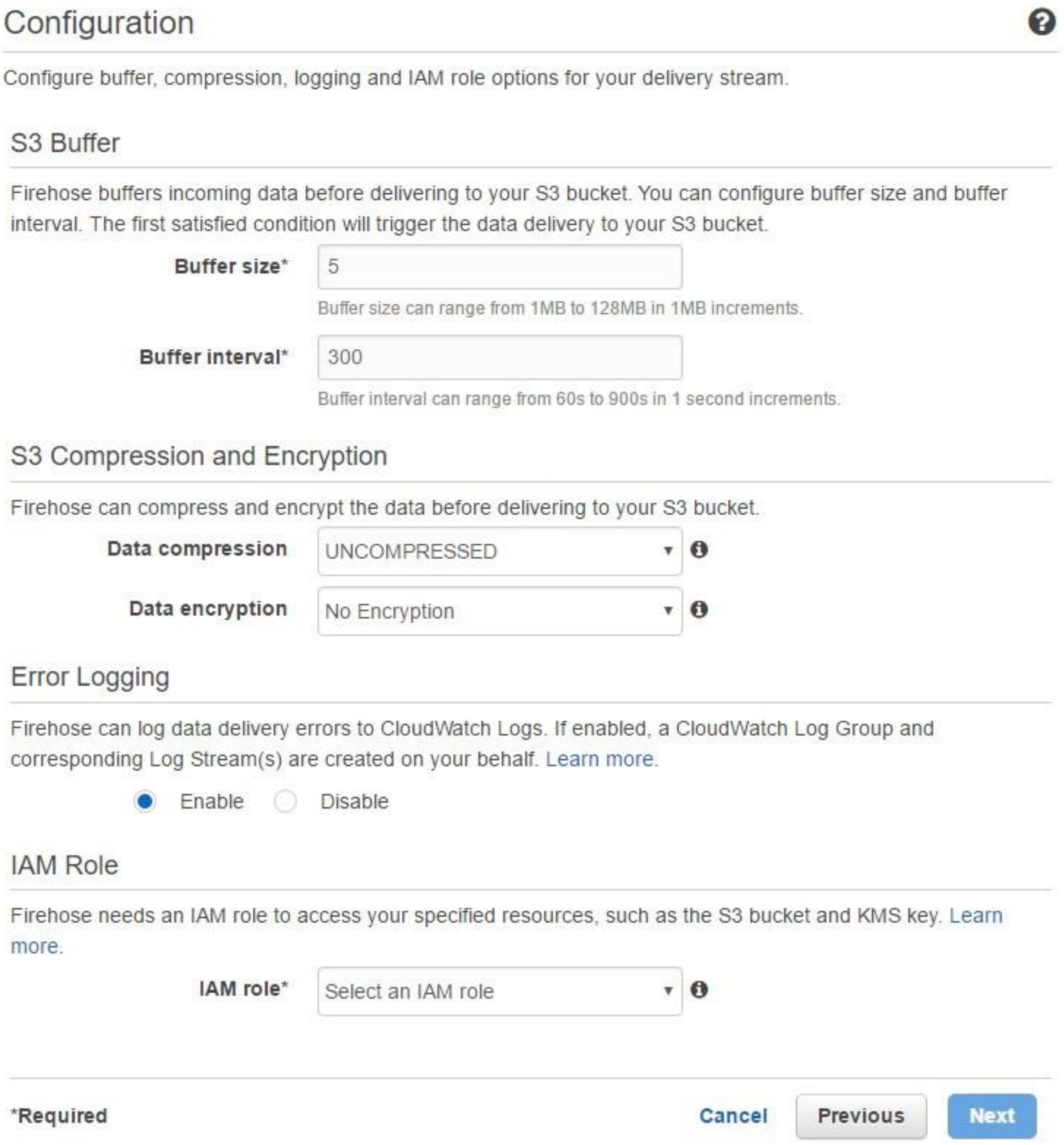
      For **Bucket name**, use a unique name. You will not need to use the name elsewhere in this tutorial. However, Amazon S3 bucket names are required to be globally unique.

      For **Region**, choose **US Standard**. Choose **Create Bucket**.

   d. For **S3 prefix**, leave it blank (default).

   e. Click **Next**.
5. On the **Configuration** screen (see below), you can leave all fields set to their default values. However, you will need to choose an IAM role so that Amazon Kinesis Firehose can write to your Amazon S3 bucket on your behalf.
   a. For **IAM role**, choose **Create/Update Existing IAM Role**.

   b. Click **Next**.



Configuration

Configure buffer, compression, logging and IAM role options for your delivery stream.

S3 Buffer

Firehose buffers incoming data before delivering to your S3 bucket. You can configure buffer size and buffer interval. The first satisfied condition will trigger the data delivery to your S3 bucket.

Buffer size* 5

Buffer size can range from 1MB to 128MB in 1MB increments.

Buffer interval* 300

Buffer interval can range from 60s to 900s in 1 second increments.

S3 Compression and Encryption

Firehose can compress and encrypt the data before delivering to your S3 bucket.

Data compression UNCOMPRESSED

Data encryption No Encryption

Error Logging

Firehose can log data delivery errors to CloudWatch Logs. If enabled, a CloudWatch Log Group and corresponding Log Stream(s) are created on your behalf. Learn more.

● Enable ○ Disable

IAM Role

Firehose needs an IAM role to access your specified resources, such as the S3 bucket and KMS key. Learn more.

IAM role* Select an IAM role

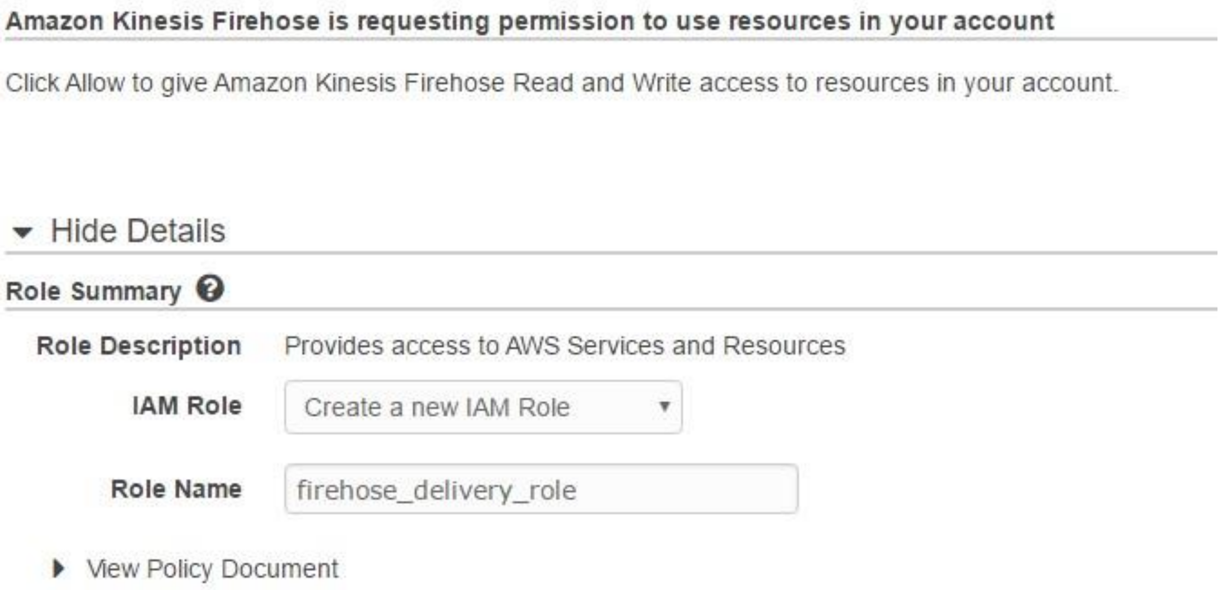
*Required

1. A new screen will open (see below).

   a. For **IAM Role**, choose **Create a new IAM Role**.

   b. For **Role Name**, enter firehose_delivery_role.

   c. Click **Next**.



Amazon Kinesis Firehose is requesting permission to use resources in your account

Click Allow to give Amazon Kinesis Firehose Read and Write access to resources in your account.

▾ Hide Details

Role Summary

Role Description Provides access to AWS Services and Resources

IAM Role Create a new IAM Role ▾

Role Name firehose_delivery_role

7. Review the details of the Amazon Kinesis Firehose delivery stream and choose **Create Delivery Stream**.

Step 3: Install and Configure the Amazon Kinesis Agent

Now that you have an Amazon Kinesis Firehose delivery stream ready to ingest your data, you can configure the EC2 instance to send the data using the Amazon Kinesis Agent software. The agent is a stand-alone Java software application that offers an easy way to collect and send data to Firehose. The agent continuously monitors a set of files and sends new data to your delivery stream. It handles file rotation, checkpointing, and retry upon failures. It delivers all of your data in a reliable, timely, and simple manner. It also emits Amazon CloudWatch metrics to help you better monitor and troubleshoot the streaming process.

The Amazon Kinesis Agent can pre-process records from monitored files before sending them to your delivery stream. It has native support for Apache access log files, which you created in Step 1. When configured, the agent will parse log files in the Apache Common Log format and convert each line in the file to JSON format before sending to your Firehose delivery stream, which you created in Step 2.

1. To install the agent, see Download and Install the Agent.[11]
2. For detailed instructions on how to configure the agent to process and send log data to your Amazon Kinesis Firehose delivery stream, see Configure and Start the Agent.[12]

   To configure the agent for this tutorial, modify the configuration file located at **/etc/aws-kinesis/agent.json** using the template below.
   Replace full-path-to-log-file with a file pattern that represents the path to your log files and a wildcard if you have multiple log files with the same naming convention. For example, it might look similar to:
   “/var/log/httpd-access.log*”. The value will be different, depending on your use case. Replace name-of-delivery-stream with the name of the Firehose delivery stream you created in Step 2.

```
{
 "cloudwatch.endpoint": "monitoring.us-east-1.amazonaws.com",
 "cloudwatch.emitMetrics": true,
 "firehose.endpoint": "firehose.us-east-1.amazonaws.com",
 "flows": [
   {
     "filePattern": "full-path-to-log-file",
     "deliveryStream": "name-of-delivery-stream",
     "dataProcessingOptions": [
     {
          "initialPostion": "START_OF_FILE",
          "maxBufferAgeMillis":"2000",
          "optionName": "LOGTOJSON",
          "logFormat": "COMBINEDAPACHELOG"
     }]
   }
 ]
}
```

3. Start the agent manually by issuing the following command:

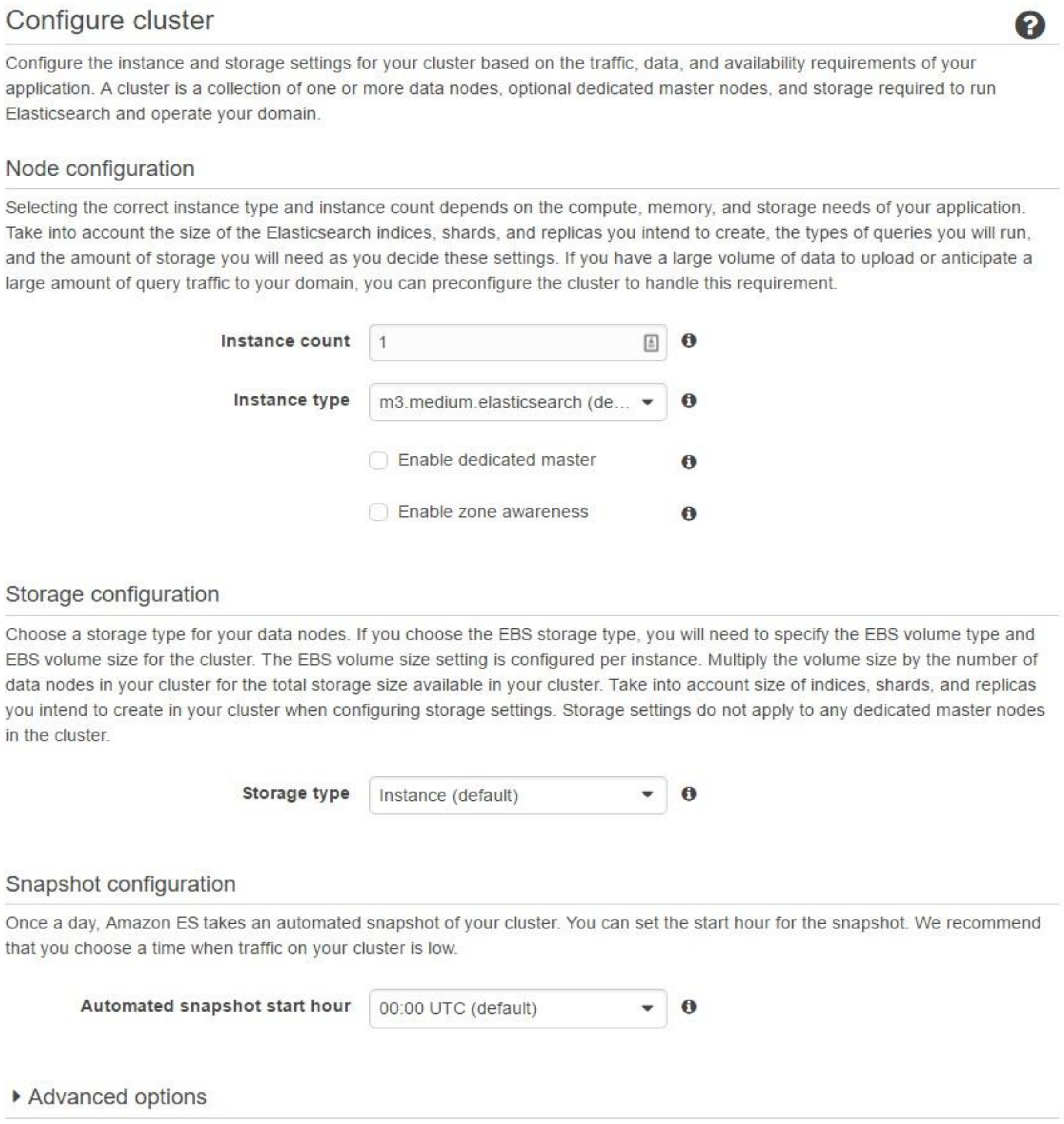
```
sudo service aws-kinesis-agent start
```

Once started, the agent will look for files in the configured location and send the records to the Firehose delivery stream.

Step 4: Create an Amazon Elasticsearch Service Domain

The data produced by this tutorial will be stored in Amazon ES for later visualization and analysis. To create the Amazon ES domain:

1. Open the Amazon ES console at https://console.aws.amazon.com/es.
2. If you have not previously created an Amazon ES Domain, choose **Get started**. Otherwise, choose **Create a new domain**.
3. On the **Define domain** screen:
    a. For **Elasticsearch domain name**, enter web-log-summary.
    b. For **Elasticsearch version**, leave it set to the default value.
    c. Click **Next**.
4. On the **Configure cluster** screen (see below), leave all settings as their default values and click **Next**.



Configure cluster

Configure the instance and storage settings for your cluster based on the traffic, data, and availability requirements of your application. A cluster is a collection of one or more data nodes, optional dedicated master nodes, and storage required to run Elasticsearch and operate your domain.

Node configuration

Selecting the correct instance type and instance count depends on the compute, memory, and storage needs of your application. Take into account the size of the Elasticsearch indices, shards, and replicas you intend to create, the types of queries you will run, and the amount of storage you will need as you decide these settings. If you have a large volume of data to upload or anticipate a large amount of query traffic to your domain, you can preconfigure the cluster to handle this requirement.

Instance count: 1

Instance type: m3.medium.elasticsearch (de…

□ Enable dedicated master

□ Enable zone awareness

Storage configuration

Choose a storage type for your data nodes. If you choose the EBS storage type, you will need to specify the EBS volume type and EBS volume size for the cluster. The EBS volume size setting is configured per instance. Multiply the volume size by the number of data nodes in your cluster for the total storage size available in your cluster. Take into account size of indices, shards, and replicas you intend to create in your cluster when configuring storage settings. Storage settings do not apply to any dedicated master nodes in the cluster.

Storage type: Instance (default)

Snapshot configuration

Once a day, Amazon ES takes an automated snapshot of your cluster. You can set the start hour for the snapshot. We recommend that you choose a time when traffic on your cluster is low.

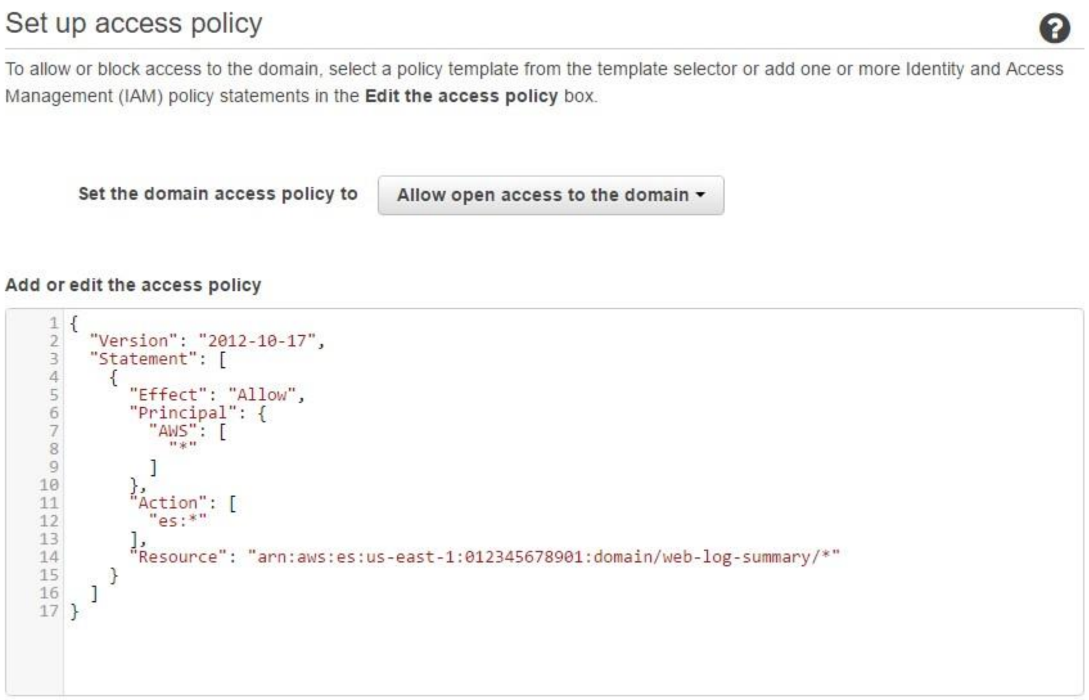
Automated snapshot start hour: 00:00 UTC (default)

5. On the **Set up access policy** screen (see below):

   a. For **Set the domain access policy to**, choose **Allow open access to the domain**.

      Note: This is not a recommended setting for production Amazon ES domains. You should terminate this Amazon ES domain after completing the tutorial, or apply a more restrictive policy.

   b. Click **Next**.



Set up access policy

To allow or block access to the domain, select a policy template from the template selector or add one or more Identity and Access Management (IAM) policy statements in the **Edit the access policy** box.

Set the domain access policy to Allow open access to the domain ▾

Add or edit the access policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "*"
        ]
      },
      "Action": [
        "es:*"
      ],
      "Resource": "arn:aws:es:us-east-1:012345678901:domain/web-log-summary/*"
    }
  ]
}
```

6. Review the details for the Amazon ES domain and click **Confirm and create**. It will take approximately 10 minutes for the Amazon ES domain to be created. While the domain is being created, proceed with the remainder of this guide.

6. Contribution Summary

Step 1: Set Up Prerequisites (Ayush Kumar Singh)

Step 2: Create an Amazon Kinesis Firehose Delivery Stream(Ishan Rathi)

Step 3: Install and Configure the Amazon Kinesis Agent (Adesh Kumar)

Step 4: Create an Amazon Elasticsearch Service Domain(Ishan Rathi)

7. Additional References

[1] https://aws.amazon.com/ec2/pricing/

[2] https://aws.amazon.com/kinesis/firehose/pricing/

[3] https://aws.amazon.com/s3/pricing/

[4] https://aws.amazon.com/free/

[5] https://aws.amazon.com/kinesis/analytics/pricing/

[6] https://aws.amazon.com/elasticsearch-service/pricing/

[7] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.ht ml

[8] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-foramazon-ec2.html

[9] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstances.html

[10] https://github.com/kiritbasu/Fake-Apache-Log-Generator

[11] http://docs.aws.amazon.com/firehose/latest/dev/writing-withagents.html#download-install

[12] http://docs.aws.amazon.com/firehose/latest/dev/writing-withagents.html#config-start

[13] https://www.elastic.co/products/kibana

[14] http://docs.aws.amazon.com/kinesisanalytics/latest/dev/how-it-works.html

[15] http://docs.aws.amazon.com/kinesisanalytics/latest/dev/streaming-sqlconcepts.html

http://docs.aws.amazon.com/kinesisanalytics/latest/dev/example-apps.html