**😊 Kinesis Data Stream and Kinesis Agent**

Amazon Kinesis Agent is a stand-alone software application designed to collect and stream data from your servers to Amazon Kinesis services, such as Kinesis Data Streams and Kinesis Data Firehose. It is typically installed on servers (such as EC2 instances) and used to continuously monitor and forward log files, metrics, or other data streams to AWS for real-time processing, storage, or analysis.

**Key Features:**

1. **Data Collection**: Kinesis Agent can monitor specified files or directories on your server and stream the data directly to Kinesis services. It’s commonly used for capturing log data, application metrics, or any custom data generated by your systems.
2. **Seamless Integration with Kinesis**: The agent natively integrates with Kinesis Data Streams and Kinesis Data Firehose, making it easy to stream collected data to those services without writing additional custom code.
3. **Automatic Retries and Buffering**: Kinesis Agent automatically retries failed data uploads and buffers data in case of network failures, ensuring reliability and minimizing data loss.
4. **Easy Configuration**: The agent is configured using a JSON configuration file where you specify the files to monitor, the destination (Kinesis Data Streams or Firehose), and other options like data transformation and compression.
5. **Cross-Platform**: It can be installed on both Linux and Windows systems.

**Common Use Cases:**

* **Log File Streaming**: Continuously capture and stream log files (e.g., web server logs, application logs) for real-time analysis or storage.
* **Metric Streaming**: Send system or application metrics to Kinesis for monitoring and alerting.
* **Data Lake Ingestion**: Stream collected data directly into S3 via Kinesis Firehose, helping build a data lake.

**Benefits:**

* **Simplicity**: No need to write custom code to forward data; the agent does it for you.
* **Reliability**: Automatic retry, buffering, and checkpointing ensure robust data collection.
* **Real-Time**: Streams data in real-time for immediate processing and analysis.

Kinesis Agent simplifies the process of collecting, processing, and analyzing streaming data from your servers, enabling real-time insights and analytics.

**In this exercise, the goal is to host a sample website on an EC2 instance and capture real-time logs generated by user interactions. These logs are collected using Kinesis Data Streams, delivered to S3 using Kinesis Data Firehose, and stored for later analysis.**

**You start by deploying a web server on EC2, creating an IAM role to allow access to Kinesis and S3, and configuring security settings. After launching the web server and hosting a website, logs generated from website activity are captured from the server. To stream these logs, you install the Kinesis agent on the EC2 instance, configure it, and push the logs to a Kinesis Data Stream. From there, the logs are delivered to an S3 bucket using Firehose, where they can be stored and analyzed.**

**The end goal is to demonstrate how to collect and store real-time logs generated by a website in S3, leveraging Kinesis services to manage and process the streaming data.**

**😄 To begin with the Lab:**

1. In this lab, we are going to host a sample website. Now, once we host this sample website on a Web server, what we will do is whatever the logs that we created in real-time, we are going to use those logs and then we are going to store them into an S3 bucket. Now to collect the data in real-time, we'll make use of the kinesis data stream and once we stream the real-time data, we'll ensure that using the firehose, we are going to deliver the collected logs into the S3 bucket.
2. First, we need to deploy the Web Server on an EC2 instance. But before that, we need an IAM role that we can attach to our EC2 instance. So, go to IAM and create a JSON policy using the below code. After that go to Roles and create a role for EC2 using the same policy.

**{**

**"Version": "2012-10-17",**

**"Statement": [**

**{**

**"Sid": "kinesisAccess",**

**"Effect": "Allow",**

**"Action": [**

**"kinesis:\*"**

**],**

**"Resource": "\*",**

**"Condition": {**

**"StringEquals": {**

**"aws:RequestedRegion": "us-east-1"**

**}**

**}**

**},**

**{**

**"Effect": "Allow",**

**"Action": [**

**"s3:\*"**

**],**

**"Resource": "arn:aws:s3:::\*"**

**}**

**]**

**}**

1. Once your role has been created then come to EC2 and click on create instance. Give it a name and choose Amazon Linux 2 as your AMI.

A screenshot of a computer program

Description automatically generated

1. Choose your instance type as t2.micro and choose your key pair.

**A screenshot of a computer

Description automatically generated**

1. After that in the network settings for the Security Group add an inbound rule for HTTP from everywhere.
2. Then expand the Advanced details and choose your IAM role, scroll down to the bottom.

A screenshot of a computer

Description automatically generated

1. In the user data paste the below script and create your instance.

**#!/bin/bash**

**sudo -s**

**yum update -y**

**sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2**

**sudo yum install -y httpd mariadb-server**

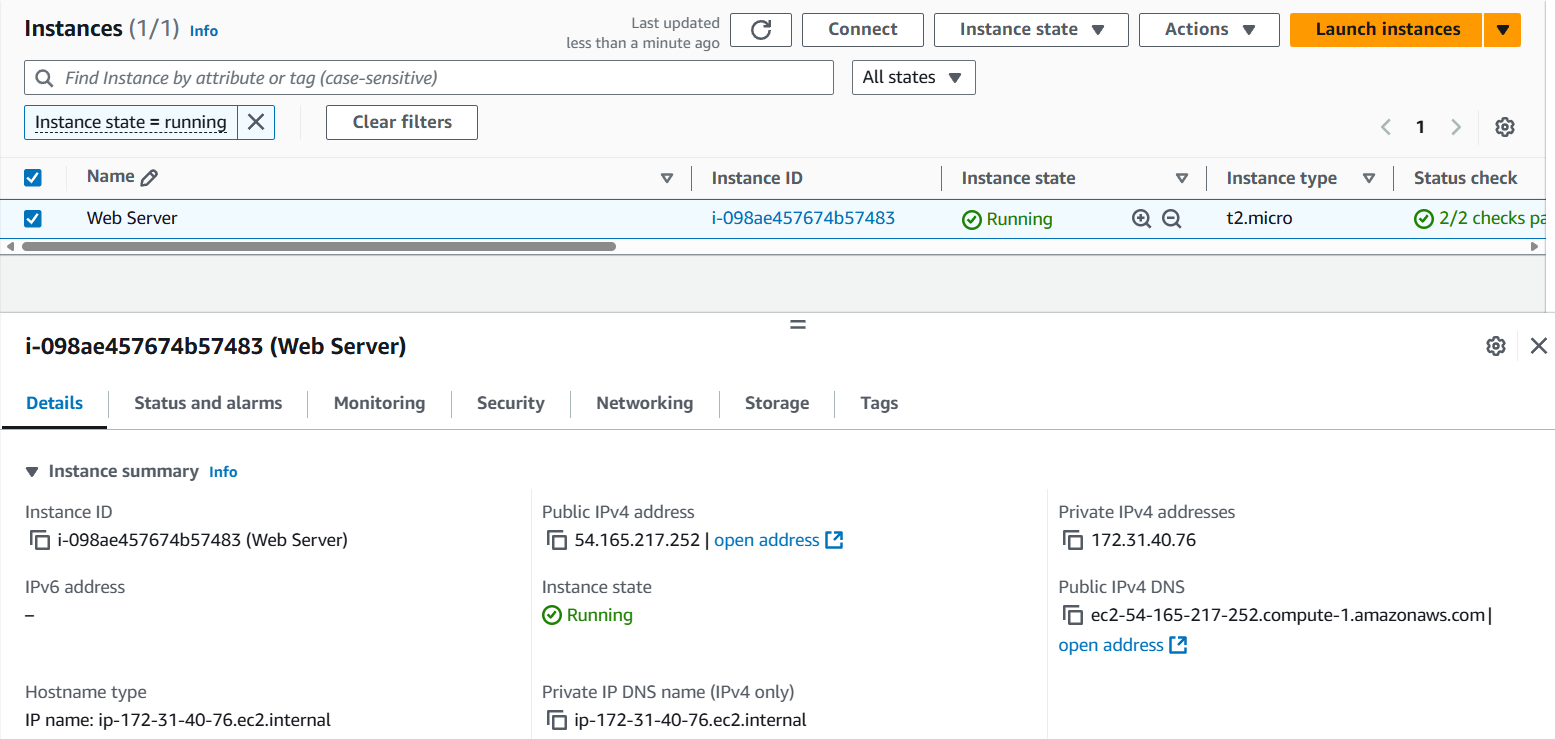
**sudo systemctl start httpd**

**sudo systemctl enable httpd**

A screenshot of a computer program

Description automatically generated

1. Once your instance is launched wait for it to get initialized and then copy its public IP address and paste it into a new tab. You will see a web server on it.



A screenshot of a computer

Description automatically generated

1. After that we need to connect with our instance. So, select your instance and click on connect then choose EC2 instance connect and click on connect.

A screenshot of a computer

Description automatically generated

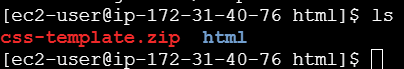
1. Once your instance is connected then you need to run some commands shown below.
2. The first command will take you to the location then the second command will let you download the zip file of a website. After that you will unzip that file, and the last command is to list the objects present.

**cd  /var/www/html**

**sudo wget https://www.dropbox.com/s/nw5bqgym23n93lf/css-template.zip**

**sudo unzip css-template.zip**

**ls**

****

1. Now if you append your IP with **/html** you will see a proper website hosted.

A screenshot of a computer

Description automatically generated

1. Now if go through this website and do some activities then some logs will be generated which you can see using the below commands. Below you can see the logs generated.
2. So, whatever the click or whatever the activities that we'll be doing on this website, each of the activities will generate a log and this log is being captured at this location.

**sudo su**

**cd /var/log/httpd/**

**tail -10 access\_log**

**A computer screen with many small colored dots

Description automatically generated with medium confidence**

1. So, to store these logs continuously we must change the permission of the HTTPD folder so that the file will be in a readable, writable, and executable mode by the EC2 user. Now we need to run the commands given below.
2. After running the below commands, you need to connect with your user again.

**groupadd httpd**

**usermod -a -G httpd ec2-user**

**exit**

A screen shot of a computer screen

Description automatically generated

1. Now after re-connecting if you call groups then you will be able to see the httpd group.

A black background with white text

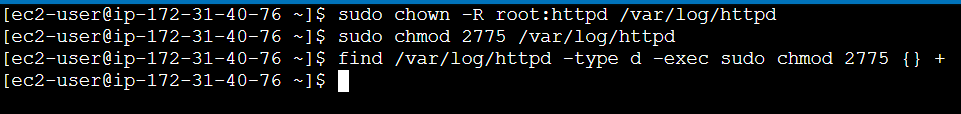
Description automatically generated

1. Now we are going to change the group ownership of the **var log Httpd directory** as well as its contents onto our Httpd group. For that you need to run the below commands.
2. Now, once we change the group ownership, the next thing that we’re going to do is we are going to change the directory permission of the var log httpd as well as its sub-directory to the ad group. And I'm going to write the permissions and set the group ID on the subdirectories which will be created in the future.

**sudo chown -R root:httpd /var/log/httpd**

**sudo chmod 2775 /var/log/httpd**

**find /var/log/httpd -type d -exec sudo chmod 2775 {} +**

****

1. After that we are going to create our Kinesis Data Stream, in your AWS Console search for Kinesis Data Stream and choose to create one.
2. Here just give it a name keep everything to default and create your data stream.

A screenshot of a computer

Description automatically generated

1. Once your data stream is created, then you need to go to configuration and enable the Server-side encryption.

A screenshot of a computer

Description automatically generated

1. Now we are going to create an S3 bucket by enabling bucket versioning.
2. Then come to Kinesis Data Firehose and click on Create. Choose the source as Data streams and the destination as Amazon S3. In the source setting choose you data stream.

A screenshot of a computer

Description automatically generated

1. Then scroll down to destination settings and choose your S3 bucket.

A screenshot of a computer

Description automatically generated

1. In the end, expand the buffer hints option and choose buffer size as 1 MiB and the buffer interval as 60 seconds.

A screenshot of a computer error

Description automatically generated

1. So, to push the log from my EC2 instance to my kinesis data stream, we will have to configure a kinesis agent which is responsible for the collection of data set and sending it to the data stream.
2. Come back to EC2 and connect your instance. Then run the command given below. So, this command will install the Kinesis agent on your Instance.

**sudo yum install –y** [**https://s3.amazonaws.com/streaming-data-agent/aws-kinesis-agent-latest.amzn2.noarch.rpm**](https://s3.amazonaws.com/streaming-data-agent/aws-kinesis-agent-latest.amzn2.noarch.rpm)

1. Once this agent is installed then you need to configure this agent. So, to do that we need to run the below command and edit the agent.json file. Run the command and you will see the data in this file you need to delete this data and write the new data in it.

**sudo nano /etc/aws-kinesis/agent.json**

A screen shot of a computer

Description automatically generated

1. Use the below script in the agent.json file. Here you need to create a user with administrator privileges and then create the access and secret access key for the user. Then change the Kinesis stream name with yours and paste this script.

**{**

**"cloudwatch.emitMetrics": true,**

**"kinesis.endpoint": "",**

**"firehose.endpoint": "",**

**"awsAccessKeyId": "AKIA4ZIQ7TEGJWSLYK4M",**

**"awsSecretAccessKey": "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",**

**"flows": [**

**{**

**"filePattern": "/var/log/httpd/access\_log",**

**"kinesisStream": "demo",**

**"partitionKeyOption": "RANDOM"**

**}**

**]**

**}**

A computer screen with text on it

Description automatically generated

1. Now save this file and now we are going to start and stop our Kinesis agent.

**sudo service aws-kinesis-agent stop**

**sudo service aws-kinesis-agent start**

A screen shot of a computer

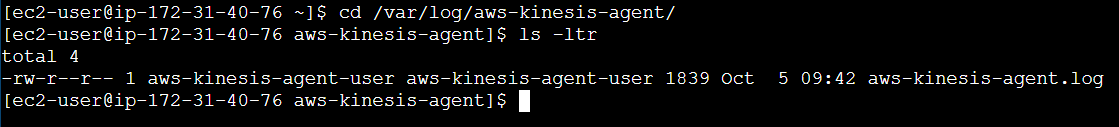
Description automatically generated

1. Now run the below commands to see the logs.

**cd /var/log/aws-kinesis-agent/**

**ls -ltr**

**head -10 aws-kinesis-agent.log**



A black background with many small colored lines

Description automatically generated with medium confidence

1. The last thing we need to do is go to the Web Site and then do some random activities so that some logs get stored in our S3 bucket.
2. After some time if you go to Data stream and Firehose, open monitoring here you will see some spike in the metrics.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Also, in your S3 bucket you will see the records which you can download and view in your Notepad.

A screenshot of a computer

Description automatically generated

1. Once you are done then delete all your resources. Start with the EC2 instance, then the S3 bucket after that delete the Kinesis data stream and Firehose.