



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": [  
                "logs:CreateLogStream",  
                "logs:PutLogEvents"  
            ]  
        }  
    ]  
}
```

```

    "S3:""
],
"Resource": "arn:aws:s3:::*"
}
]
}

```

- 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.

Recents Quick Start

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type		Free tier eligible
ami-00f251754ac5da7f0 (64-bit (x86)) / ami-033e2e260e34da7cc (64-bit (Arm))		
Virtualization: hvm ENA enabled: true Root device type: ebs		

Description

Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.

Architecture	AMI ID	Username	
64-bit (x86)	ami-00f251754ac5da7f0	ec2-user	

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

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro	Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true On-Demand Windows base pricing: 0.0162 USD per Hour On-Demand SUSE base pricing: 0.0116 USD per Hour On-Demand RHEL base pricing: 0.026 USD per Hour On-Demand Linux base pricing: 0.0116 USD per Hour	<input checked="" type="radio"/> All generations
Compare instance types	

Additional costs apply for AMIs with pre-installed software

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

The screenshot shows the 'Advanced details' section of the AWS Lambda configuration interface. It includes fields for 'Domain join directory', 'IAM instance profile', and 'Hostname type'. Each field has an 'Info' link and a dropdown menu. To the right of each dropdown are 'Create new' buttons: 'Create new directory' for domain join, 'Create new IAM profile' for instance profile, and 'Create new' for hostname type.

▼ Advanced details [Info](#)

Domain join directory | [Info](#)

Select ▾ [Create new directory](#)

IAM instance profile | [Info](#)

ec2-ML-Role
arn:aws:iam::878893308172:instance-profile/ec2-ML-Role ▾ [Create new IAM profile](#)

Hostname type | [Info](#)

IP name ▾

7. 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
```

User data - optional | [Info](#)

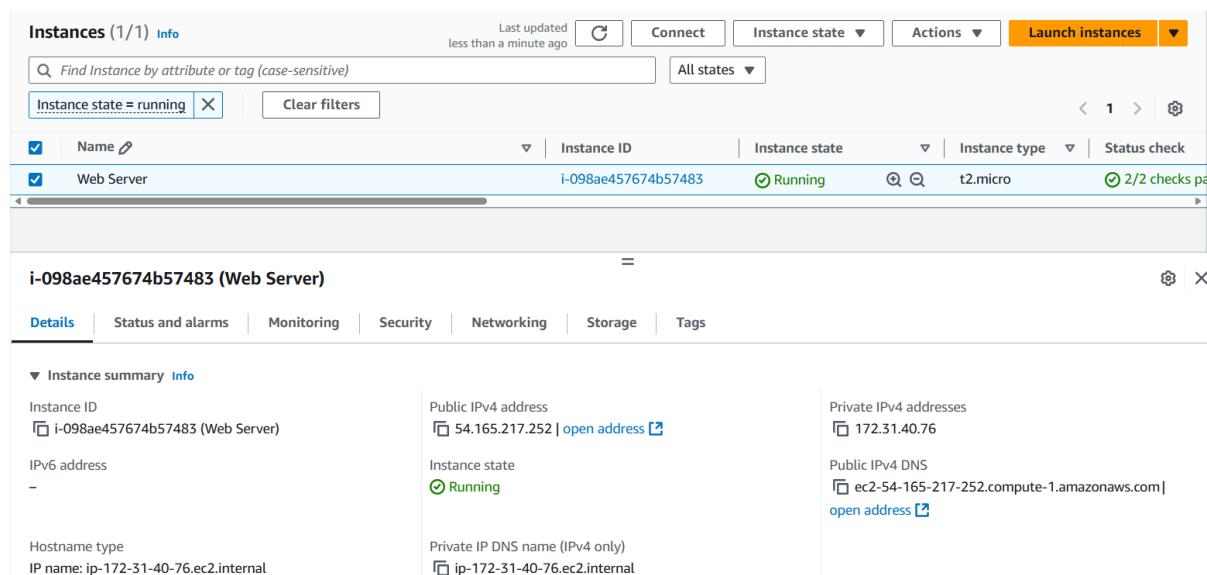
Upload a file with your user data or enter it in the field.

 Choose file

```
#!/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
```

- 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.



The screenshot shows the AWS CloudWatch Instances console. At the top, there's a search bar with placeholder text "Find Instance by attribute or tag (case-sensitive)" and a dropdown menu set to "All states". Below the search bar, there are two filter buttons: "Instance state = running" and "Clear filters". The main table lists one instance:

Name	Instance ID	Instance state	Instance type	Status check
<input checked="" type="checkbox"/> Web Server	i-098ae457674b57483	Running	t2.micro	2/2 checks passed

Below the table, the instance details are shown under the heading "i-098ae457674b57483 (Web Server)". The "Details" tab is selected, showing the following information:

- Instance summary**:
 - Instance ID: i-098ae457674b57483 (Web Server)
 - IPv6 address: -
 - Hostname type: IP name: ip-172-31-40-76.ec2.internal
- Public IPv4 address**: 54.165.217.252 | [open address](#)
- Private IPv4 addresses**: 172.31.40.76
- Public IPv4 DNS name**: ec2-54-165-217-252.compute-1.amazonaws.com | [open address](#)
- Private IP DNS name (IPv4 only)**: ip-172-31-40-76.ec2.internal



This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory /var/www/html/. Note that until you do so, people visiting your website will see this page and not your content. To prevent this page from ever being used, follow the instructions in the file /etc/httpd/conf.d/welcome.conf.

You are free to use the image below on web sites powered by the Apache HTTP Server.



- 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.

Connect to instance Info

Connect to your instance i-098ae457674b57483 (Web Server) using any of these options

EC2 Instance Connect

Session Manager

SSH client

EC2 serial console



Port 22 (SSH) is open to all IPv4 addresses

Port 22 (SSH) is currently open to all IPv4 addresses, indicated by **0.0.0.0/0** in the inbound rule in [your security group](#). For increased security, consider restricting access to only the EC2 Instance Connect service IP addresses for your Region: 18.206.107.24/29. [Learn more](#).

Instance ID

[i-098ae457674b57483 \(Web Server\)](#)

- Once your instance is connected then you need to run some commands shown below.
- 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
```

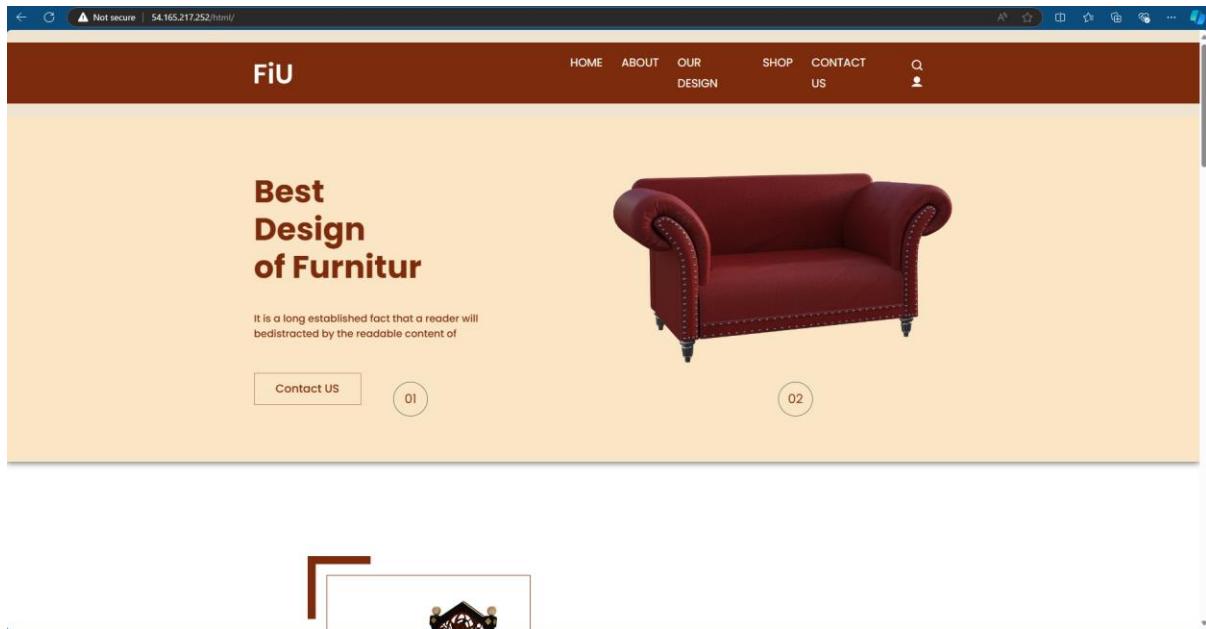
```
sudo wget https://www.dropbox.com/s/nw5bqgym23n93lf/css-template.zip
```

```
sudo unzip css-template.zip
```

```
ls
```

```
[ec2-user@ip-172-31-40-76 html]$ ls
css-template.zip  html
[ec2-user@ip-172-31-40-76 html]$ █
```

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



13. 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.
14. 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
```

```
[ec2-user@ip-172-31-40-76 html]$ sudo su
[root@ip-172-31-40-76 html]# cd /var/log/httpd/
[root@ip-172-31-40-76 httpd]# tail -10 access_log
103.226.202.63 - - [05/Oct/2024:08:31:27 +0000] "GET /html/about.html HTTP/1.1" 200 7466 "http://54.165.217.252/html/index.html" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 Safari/537.36 Edg/129.0.0.0"
103.226.202.63 - - [05/Oct/2024:08:31:27 +0000] "GET /html/images/banner-bg.png HTTP/1.1" 404 196 "http://54.165.217.252/html/css/style.css" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 Safari/537.36 Edg/129.0.0.0"
103.226.202.63 - - [05/Oct/2024:08:31:27 +0000] "GET /html/images/blog-bg.png HTTP/1.1" 404 196 "http://54.165.217.252/html/css/style.css" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 Safari/537.36 Edg/129.0.0.0"
103.226.202.63 - - [05/Oct/2024:08:31:28 +0000] "GET /html/css/owl.theme.default.min.css HTTP/1.1" 404 196 "http://54.165.217.252/html/about.html" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 Safari/537.36 Edg/129.0.0.0"
103.226.202.63 - - [05/Oct/2024:08:31:28 +0000] "GET /html/images/banner-bg.png HTTP/1.1" 404 196 "http://54.165.217.252/html/css/style.css" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 Safari/537.36 Edg/129.0.0.0"
103.226.202.63 - - [05/Oct/2024:08:31:28 +0000] "GET /html/contact.html HTTP/1.1" 200 8100 "http://54.165.217.252/html/about.html" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 Safari/537.36 Edg/129.0.0.0"
103.226.202.63 - - [05/Oct/2024:08:31:29 +0000] "GET /html/shop.html HTTP/1.1" 200 7311 "http://54.165.217.252/html/about.html" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 Safari/537.36 Edg/129.0.0.0"
103.226.202.63 - - [05/Oct/2024:08:31:29 +0000] "GET /html/css/owl.theme.default.min.css HTTP/1.1" 404 196 "http://54.165.217.252/html/contact.html" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 Safari/537.36 Edg/129.0.0.0"
103.226.202.63 - - [05/Oct/2024:08:31:29 +0000] "GET /html/images/banner-bg.png HTTP/1.1" 404 196 "http://54.165.217.252/html/css/style.css" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 Safari/537.36 Edg/129.0.0.0"
[root@ip-172-31-40-76 httpd]#
```

15. 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.
16. After running the below commands, you need to connect with your user again.

```
groupadd httpd
usermod -a -G httpd ec2-user
exit
```

```
[root@ip-172-31-40-76 httpd]# groupadd httpd
[root@ip-172-31-40-76 httpd]# usermod -a -G httpd ec2-user
[root@ip-172-31-40-76 httpd]# exit
exit
[ec2-user@ip-172-31-40-76 html]$ exit
logout
```

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

```
[ec2-user@ip-172-31-40-76 ~]$ groups
ec2-user adm wheel systemd-journal httpd
[ec2-user@ip-172-31-40-76 ~]$ █
```

18. 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.
19. 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 {} +
```

```
[ec2-user@ip-172-31-40-76 ~]$ sudo chown -R root:httpd /var/log/httpd
[ec2-user@ip-172-31-40-76 ~]$ sudo chmod 2775 /var/log/httpd
[ec2-user@ip-172-31-40-76 ~]$ find /var/log/httpd -type d -exec sudo chmod 2775 {} +
```

20. 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.
21. Here just give it a name keep everything to default and create your data stream.

Data stream configuration

Data stream name

DataStream

Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens and periods.

Data stream capacity Info

Capacity mode

On-demand

Use this mode when your data stream's throughput requirements are unpredictable and variable. With on-demand mode, your data stream's capacity scales automatically.

Provisioned

Use provisioned mode when you can reliably estimate throughput requirements of your data stream. With provisioned mode, your data stream's capacity is fixed.

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

[Amazon Kinesis](#) > [Data streams](#) > [DataStream](#) > [Edit encryption](#)

Edit encryption for DataStream

Encryption Info

Enable server-side encryption

Kinesis Data Stream uses AWS Key Management Service (KMS) to encrypt your data. You can choose the AWS managed customer master key (CMK) to encrypt your data or specify a customer-managed CMK.

Use AWS managed CMK

The AWS managed CMK (aws/kinesis) in your account is created, managed, and used on your behalf by Kinesis Data Streams.

Use customer-managed CMK

Customer-managed CMKs in your AWS account are created, owned, and managed by you.

[Cancel](#)

[Save changes](#)

23. Now we are going to create an S3 bucket by enabling bucket versioning.

24. 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.

Choose source and destination

Specify the source and the destination for your Firehose stream. You cannot change the source and destination of your Firehose stream once it has been created.

Source | [Info](#)

Amazon Kinesis Data Streams

Destination | [Info](#)

Amazon S3

Source settings

Kinesis data stream

arn:aws:kinesis:us-east-1:878893308172:stream/DataStream

[Browse](#)

[Create](#) 

Format: arn:aws:kinesis:[Region]:[AccountId]:stream/[StreamName]

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

Destination settings [Info](#)

Specify the destination settings for your Firehose stream.

S3 bucket

s3://s3-bucket-logs-1852

[Browse](#)

[Create](#) 

Format: s3://bucket

New line delimiter

You can configure your Firehose stream to add a new line delimiter between records in objects that are delivered to Amazon S3.

- Not enabled
- Enabled

Dynamic partitioning [Info](#)

Dynamic partitioning enables you to create targeted data sets by partitioning streaming S3 data based on partitioning keys. You can partition your source data with inline parsing and/or the specified AWS Lambda function. You can enable dynamic partitioning only when you create a new Firehose stream. You cannot enable dynamic partitioning for an existing Firehose stream. Enabling dynamic partitioning incurs additional costs per GiB of partitioned data. For more information, see [Amazon Data Firehose pricing](#).

- Not enabled
- Enabled

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

▼ Buffer hints, compression, file extension and encryption

The fields below are pre-populated with the recommended default values for S3. Pricing may vary depending on storage and request costs.

S3 buffer hints

Amazon Data Firehose buffers incoming records before delivering them to your S3 bucket. Record delivery is triggered once the value of either of the specified buffering hints is reached.

Buffer size

The higher buffer size may be lower in cost with higher latency. The lower buffer size will be faster in delivery with higher cost and less latency.

MiB

Minimum: 1 MiB, maximum: 128 MiB. Recommended: 5 MiB.

Buffer interval

The higher interval allows more time to collect data and the size of data may be bigger. The lower interval sends the data more frequently and may be more advantageous when looking at shorter cycles of data activity.

seconds

Minimum: 0 seconds, maximum: 900 seconds. Recommended: 300 seconds.

27. 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.
28. 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>

29. 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

```
GNU nano 2.9.8                               /etc/aws-kinesis/agent.json

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

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

30. 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":
  "XNX8fJdX8dWUGMMhZ2iKD8jt2WGuwnCrUlNq0OnY",
  "flows": [
    {
      "filePattern": "/var/log/httpd/access_log",
      "kinesisStream": "demo",
      "partitionKeyOption": "RANDOM"
    }
  ]
}
```

GNU nano 2.9.8

/etc/aws-kinesis/agent.json

```
{
  "cloudwatch.emitMetrics": true,
  "kinesis.endpoint": "",
  "firehose.endpoint": "",
  "awsAccessKeyId": "AKIA4ZIQ7TEGJWSLYK4M",
  "awsSecretAccessKey": "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",
  "flows": [
    {
      "filePattern": "/var/log/httpd/access_log",
      "kinesisStream": "DataStream",
      "partitionKeyOption": "RANDOM"
    }
  ]
}
```

31. 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
```

```
[ec2-user@ip-172-31-40-76 ~]$ sudo service aws-kinesis-agent stop
Stopping aws-kinesis-agent (via systemctl): [ OK ]
[ec2-user@ip-172-31-40-76 ~]$ sudo service aws-kinesis-agent start
Starting aws-kinesis-agent (via systemctl): [ OK ]
[ec2-user@ip-172-31-40-76 ~]$ █
```

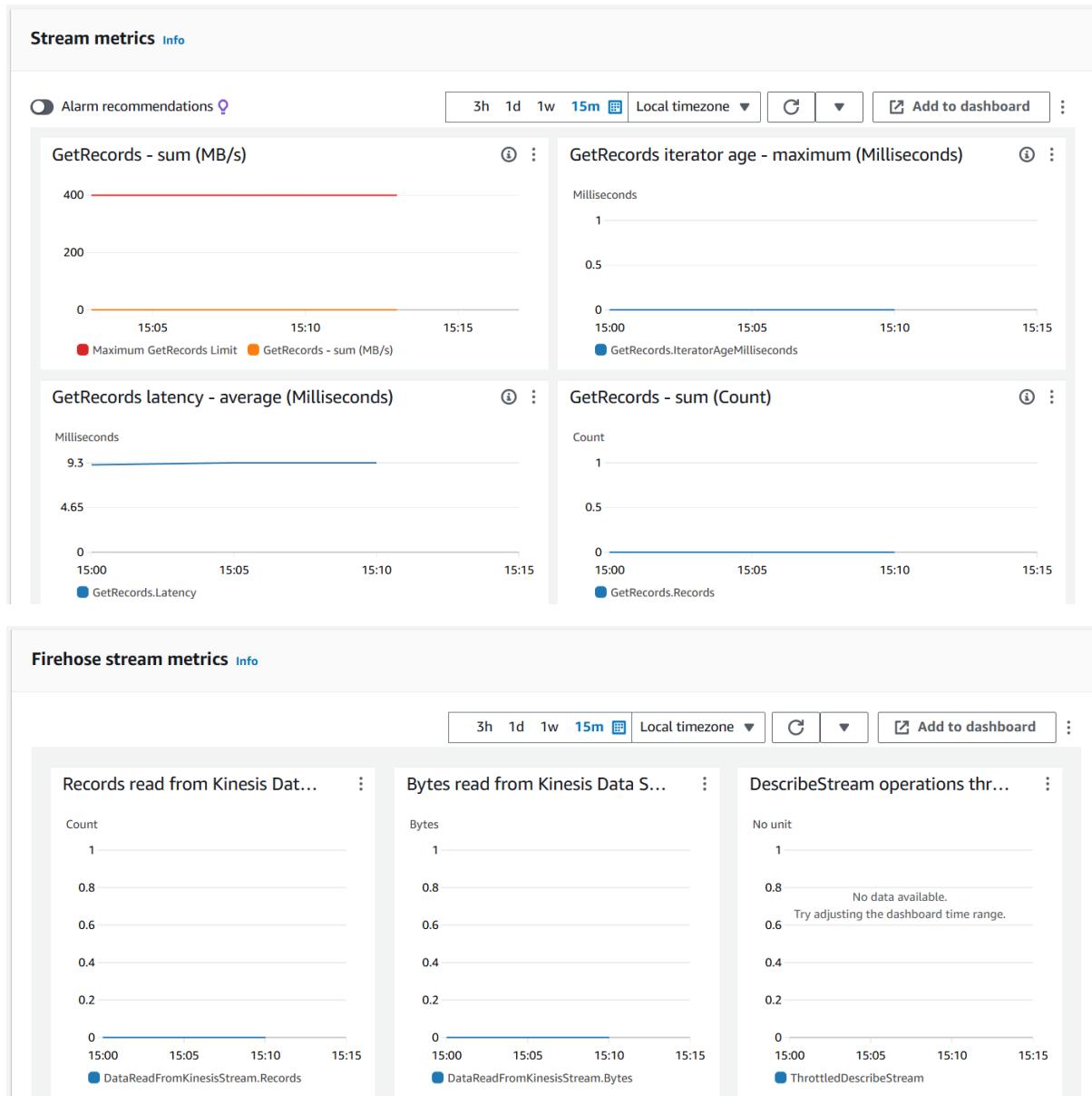
32. Now run the below commands to see the logs.

```
cd /var/log/aws-kinesis-agent/
ls -ltr
head -10 aws-kinesis-agent.log
```

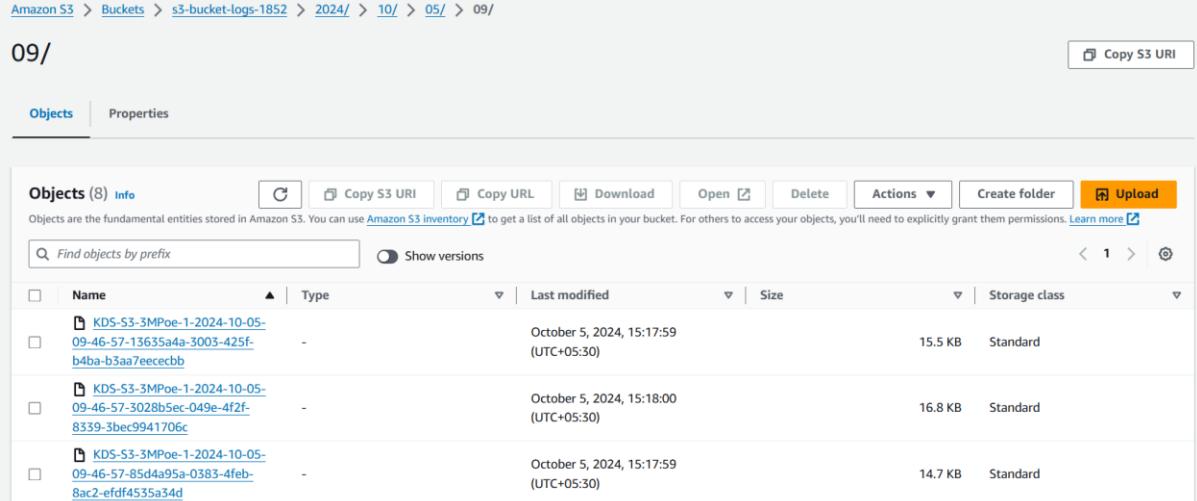
```
[ec2-user@ip-172-31-40-76 ~]$ cd /var/log/aws-kinesis-agent/
[ec2-user@ip-172-31-40-76 aws-kinesis-agent]$ ls -ltr
total 4
-rw-r--r-- 1 aws-kinesis-agent-user aws-kinesis-agent-user 1839 Oct  5 09:42 aws-kinesis-agent.log
[ec2-user@ip-172-31-40-76 aws-kinesis-agent]$
```

```
[ec2-user@ip-172-31-40-76 aws-kinesis-agent]$ head -10 aws-kinesis-agent.log
2024-10-05 09:42:15.984+0000  (main) com.amazon.kinesis.streaming.agent.Agent [INFO] Reading configuration from file: /etc/aws-kinesis/agent.json
2024-10-05 09:42:15.989+0000  (main) com.amazon.kinesis.streaming.agent.Agent [INFO] Found 1 configured flow(s)
2024-10-05 09:42:16.016+0000  (main) com.amazon.kinesis.streaming.agent.Agent [INFO] null: Agent will use up to 12 threads for sending data.
2024-10-05 09:42:16.183+0000  (main) com.amazon.kinesis.streaming.agent.tailing.checkpoints.SQLiteFileCheckpointStore [INFO] Deleted 0 old checkpoints.
2024-10-05 09:42:16.186+0000  (Agent STARTING) com.amazon.kinesis.streaming.agent.Agent [INFO] Agent: Starting up...
2024-10-05 09:42:16.190+0000  (Agent STARTING) com.amazon.kinesis.streaming.agent.Agent [INFO] Agent: Starting tailer for file kinesis:DataStream:/var/log/httpd/access_log
2024-10-05 09:42:16.221+0000  (FileTailer[kinesis:DataStream:/var/log/httpd/access_log]) com.amazon.kinesis.streaming.agent.tailing.KinesisParser [INFO] KinesisParser[Kinesis:DataStream:/var/log/httpd/access_log]: Opening /var/log/httpd/access_log for parsing.
2024-10-05 09:42:16.235+0000  (Agent STARTING) com.amazon.kinesis.streaming.agent.Agent [INFO] Agent: Startup completed in 48 ms.
2024-10-05 09:42:46.258+0000  (Agent,MetricsEmitter RUNNING) com.amazon.kinesis.streaming.agent.Agent [INFO] Agent: Progress: 0 records parsed (15571 bytes), and 0 records sent successfully to destinations. Uptime: 3007ms
2024-10-05 09:42:46.264+0000  (FileTailer[kinesis:DataStream:/var/log/httpd/access_log].MetricsEmitter RUNNING) com.amazon.kinesis.streaming.agent.tailing.FileTailer [INFO] FileTailer[kinesis:DataStream:/var/log/httpd/access_log]: Tailer Progress: Tailer has parsed 0 records (15571 bytes), transformed 0 records, skipped 0 records, and has successfully sent 0 records to destination.
[ec2-user@ip-172-31-40-76 aws-kinesis-agent]$
```

33. 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.
34. After some time if you go to Data stream and Firehose, open monitoring here you will see some spike in the metrics.



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



The screenshot shows the Amazon S3 console interface. At the top, there's a breadcrumb navigation: Amazon S3 > Buckets > s3-bucket-logs-1852 > 2024/ > 10/ > 05/ > 09/. Below the navigation is a path '09/'. On the right, there's a 'Copy S3 URI' button. The main area is titled 'Objects (8) Info'. It includes a search bar 'Find objects by prefix' and a 'Show versions' toggle. Below these are buttons for 'Actions', 'Create folder', and 'Upload'. A note says 'Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions.' There's also a 'Learn more' link. The table lists eight objects:

Name	Type	Last modified	Size	Storage class
KDS-S3-3MPoe-1-2024-10-05-09-46-57-13635a4a-3003-425fb4ba-b3aa7eececcb	-	October 5, 2024, 15:17:59 (UTC+05:30)	15.5 KB	Standard
KDS-S3-3MPoe-1-2024-10-05-09-46-57-3028b5ec-049e-4f2f-8339-3bec9941706c	-	October 5, 2024, 15:18:00 (UTC+05:30)	16.8 KB	Standard
KDS-S3-3MPoe-1-2024-10-05-09-46-57-85d4a95a-0383-4feb-8ac2-efdf4535a34d	-	October 5, 2024, 15:17:59 (UTC+05:30)	14.7 KB	Standard

36. 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.