# Lab Steps

## **Task 1: Sign in to AWS Management Console**

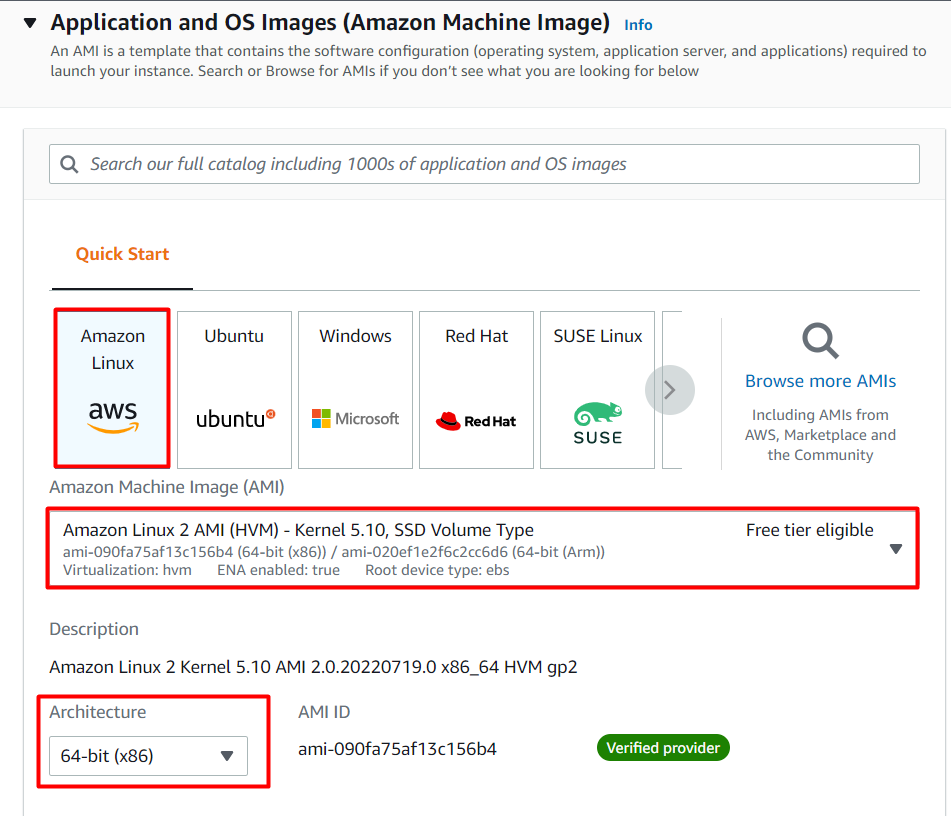
1. Click on the **Open Console** button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,
   * Leave the Account ID as default. Never edit/remove the 12 digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
   * Now copy your **User Name** and **Password** in the Lab Console to the **IAM Username and Password** in AWS Console and click on the **Sign in** button.
3. Once Signed In to the AWS Management Console, Make the default AWS Region as **US East (N. Virginia) us-east-1.**

**Note: There is no Validation function for this lab.**

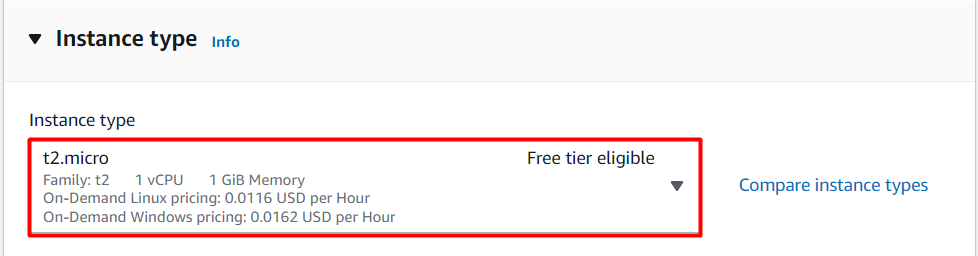
## **Task 2: Launching an EC2 Instance**

1. Make sure you are in**US East (N. Virginia) us-east-1**Region.
2. Navigate to **EC2** by clicking on the **Services**menu at the top, then click on **EC2** in the **Compute** section.
3. Navigate to **Instances**on the left panel and click on **Launch Instance**button.
4. Enter **Name** as **Demo\_Instance**
5. Select **Amazon Linux**from the **Quick Start.**

* **Choose an Amazon Machine Image (AMI):** Choose **Amazon Linux 2 AMI(HVM)** from the drop-down.

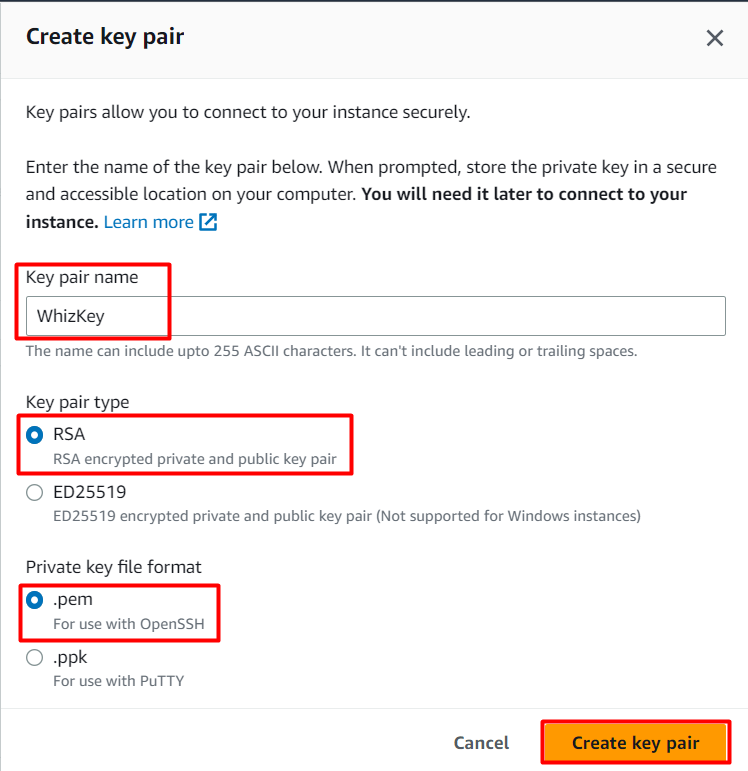


1. **Choose an Instance Type:** Select **t2.micro**

**

    7. For Key pair : Choose **Create a new key pair**

* Key Pair name : Enter **WhizKey**
* Key pair type**:**Choose **RSA**
* Private key file format: Choose **.pem**
* Click on **Create key pair** button.

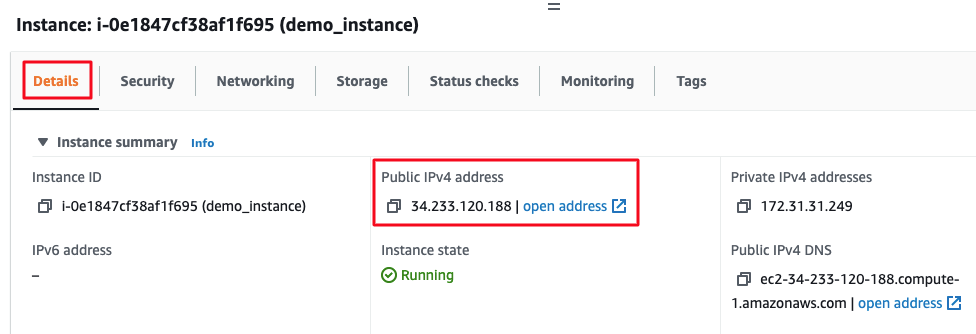


1. In Network Settings Click on **Edit** Button:
   * Auto-assign public IP: Select **Enable**
   * Select **Create new Security group**
   * Security group name : Enter **kinesis\_demo\_SG**
   * Description : Enter **Security Group to allow traffic to EC2**
   * SSH rule will already be present for you. To add HTTP:  
     + Select **Add Security rule**Button
     + Choose Type: **HTTP**
     + Source:  Select **Anywhere**
2. Under **Advanced Details,**
   * IAM Instance profile : Select **EC2\_Role\_<RANDOM\_NUMBER>**

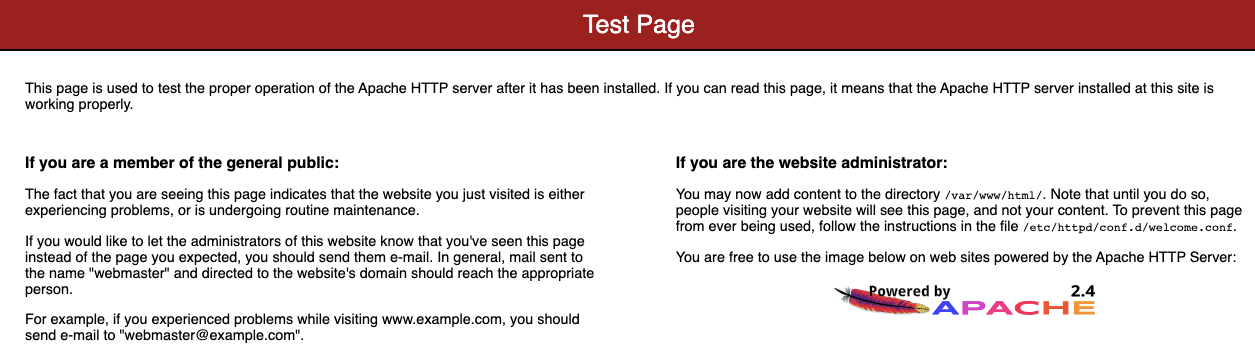
**A screenshot of a computer

Description automatically generated with medium confidence**

1. Keep Rest thing Default and Click on**Launch Instance**Button.
2. Select **View all Instances** to View Instance you Created
3. **Launch Status:** Your instances are now launching, Navigate to **Instances** page from left menu and wait the status of the EC2 Instance changes to running and health check status changes to **2/2 checks passed**
4. Select the Instance and copy the Public IPv4 address from the Details section. Note down the Public IPv4 Address of your EC2 instance. A sample is shown in the screenshot below.



1. Wait for some time before verifying the web page.
2. Paste the copied IP address in a browser window and press **[Enter]**
   * **http://**Your\_IPv4\_Address**/**
3. You should be able to see a test page. A sample is shown below.



## **Task 3: SSH into EC2 Instance**

* Please follow the steps in [SSH into EC2 Instance](https://www.whizlabs.com/labs/support-document/ssh-into-ec-instance).

## **Task 4: Host a sample website**

In this task, we will host a sample website by navigating to the HTML folder present in the var directory and then we will fetch the sample site using the wget command.

1. Switch to the root user :

sudo -s

1. Run all the updates using the yum command:

yum update -y

1. Install the LAMP server:

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

1. Install the HTTPD :

sudo yum install -y httpd mariadb-server

1. Start the HTTPD server:

sudo systemctl start httpd

1. Enable the HTTPD Server:

sudo systemctl enable httpd

1. Navigate to the HTML folder path.

cd /var/www/html

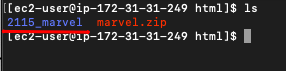
1. Let us download a sample website template. Here I am downloading a zip file from the [site](https://www.free-css.com/free-css-templates) using **wget**.

sudo wget https://www.free-css.com/assets/files/free-css-templates/download/page270/marvel.zip

1. You can check if it is downloaded in the HTML path using **ls** command.
2. Unzip the downloaded html template. Use the zip file name to unzip.

sudo unzip marvel.zip

1. Use the command **ls** to list all the files and folders present in the present working directory.

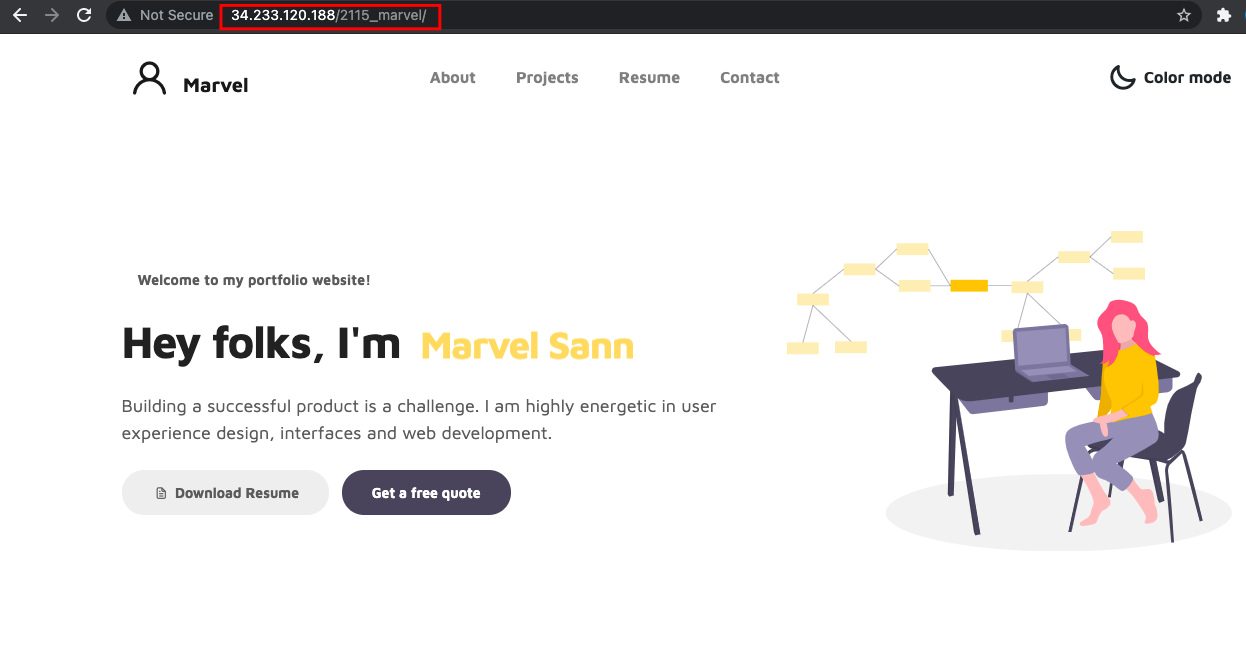


   12. You will be able to see a **zip file** and a **folder**. When we unzipped the marvel.zip, we got the folder, **2115\_marvel**.

   13. Copy the folder name to a text editor.

   14. To verify if the sample website is hosted, paste **http://IP\_Address/folder\_name** in the browser and press **[Enter]**

* http://**34.233.120.188**/**2115\_marvel**/



   15. You can see that the website is hosted successfully.

   16. The website logs will be in the path “**/var/log/httpd/access\_log**”. For each click and use of the website, the related logs will be collected and stored here.

   17. You can check the logs using the following commands

sudo su

cd /var/log/httpd/

tail -10 access\_log

## **Task 5: Set file permissions to httpd**

Now let's see how to store these continuous logs. Before proceeding, change the permission of the httpd folder, so that the file will be in readable, writable, and executable mode by ec2-user.

1. Add the **httpd** group to your EC2 instance with the command

groupadd httpd

1. Add **ec2-user** user to the **httpd** group with the command

usermod -a -G httpd ec2-user

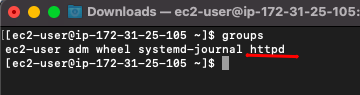
1. To refresh your permissions and include the new httpd group, log out completely with the command

exit

* + **exit** (if you are in the sudo, you have to exit twice)

1. Follow the steps to [SSH into EC2 Instance](https://www.whizlabs.com/labs/support-document/ssh-into-ec-instance).
2. Verify that the **httpd** group exists with the groups with the command

groups



1. Change the group ownership of the **/var/log/httpd** directory and its contents to the **httpd** group,

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

1. Change the directory permissions of /var/log/httpd and its subdirectories to add group write permissions and set the group ID on subdirectories created in the future,

sudo chmod 2775 /var/log/httpd

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

## **Task 6: Creating Kinesis Data Stream**

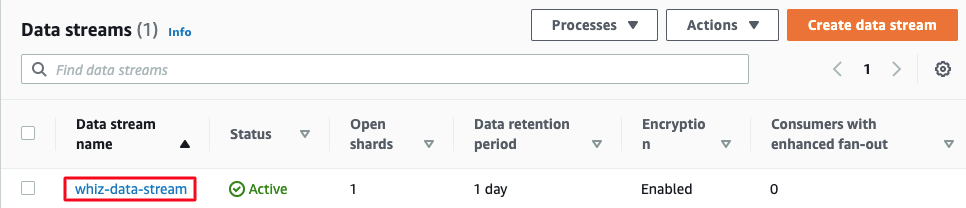
Let us create a Kinesis data stream. The logs created in the EC2 sample website will be pushed to Kinesis data stream.

1. Make sure you are inthe **US East (N. Virginia) us-east-1** Region.
2. Navigate to **Kinesis** by clicking on the **Services** menu, under the **Analytics** section.
3. Under **Get Started**, select **Kinesis Data Streams** and click on **Create data stream**.

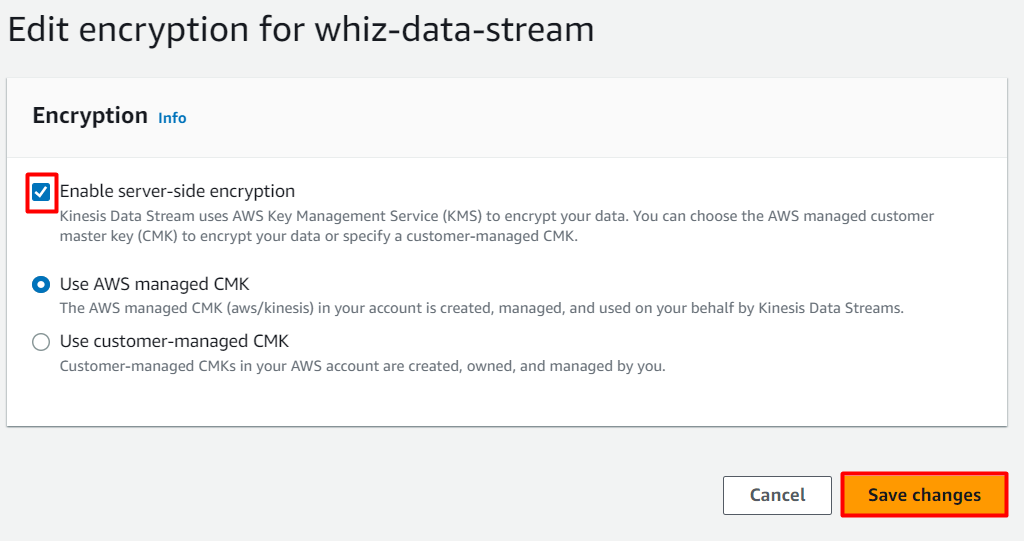
A screenshot of a computer

Description automatically generated with medium confidence

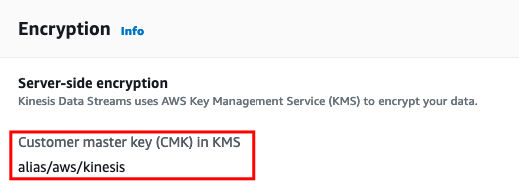
1. Under **Data stream name**, enter the **Data stream name** as **whiz-data-stream**
2. Click on **Create data stream**button.



1. Once the data stream is created, click to open it.
2. Click on the **Configuration** tab.
3. Scroll down to **Encryption** and click on **Edit**button.
4. Check **Enable server-side encryption** and use the default encryption key type, i.e **Use AWS managed CMK**.
5. Click on **Save changes**button.



1. You have used AWS KMS to encrypt your data.



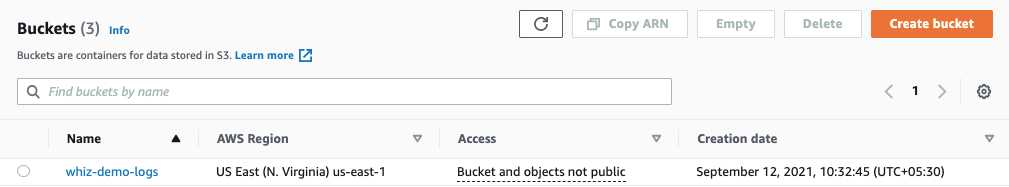
## **Task 7: Creating a S3 Bucket**

In this task, we will create an S3 bucket where we will store the data from the firehose.

1. Make sure you are inthe **US East (N. Virginia) us-east-1** Region.
2. Navigate to **S3** by clicking on the **Services** menu, under the **Storage** section.
3. Click on **Create bucket**button.
4. In the General Configuration,
   * **Bucket name**:Enter **whiz-demo-logs**
   * **Note:** S3 Bucket names are globally unique, choose a name that is available.
5. **Region**:Select **US East (N. Virginia) us-east-1** (i.e same region as the Kinesis data stream).
6. In the Default encryption,

* **Encryption key type**: Leave the key type as **Amazon S3 key (SSE-S3)**.
* **Bucket key**: Select **Enable**

1. Click on **Create bucket**button.



## **Task 8: Creating Kinesis Data Firehose**

Once the streaming service gets the data from the logs, then we need to push the data somewhere. It is not possible to post the data from the Kinesis Data Streams. So we will use Kinesis Data Firehose.

1. Make sure you are inthe **US East (N. Virginia) us-east-1** Region.
2. Navigate to **Kinesis** by clicking on the **Services** menu, under the **Analytics** section.
3. Under **Get Started**, select **Kinesis Data Firehose** and click on **Create delivery stream**.
4. Under **Choose Source and Destination**,
   * **Source**: Choose **Amazon Kinesis Data Streams**
   * **Destination**: Choose **Amazon S3**

**A picture containing text, screenshot, font, line

Description automatically generated**

     5. Under the **Source settings**,

* Click on **Browse**button.
* From the pop-up, select the Data Stream we have created earlier
* Click on **Choose**button.

    6. Under **Delivery stream name**, Enter **Delivery stream name**as **whiz-delivery-stream**

    7. Leave the **Transform and convert records** as default.

    8. Under the **Destination settings**,

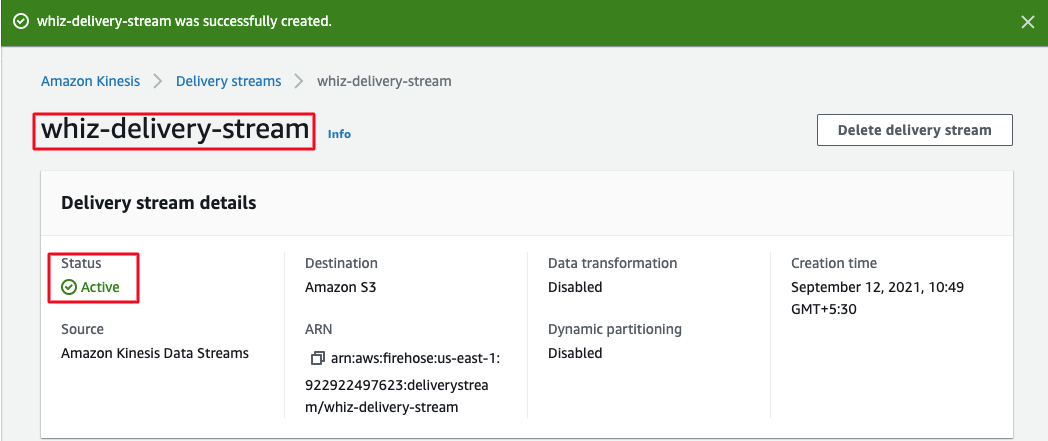
* Click on **Browse**button.
* From the pop-up, select the S3 Bucket we have created earlier, in my case, **whiz-demo-logs**
* Click on **Choose**button.

    9. Expand the **Buffer hints, compression and encryption** section. Under the **Buffer interval**, make it to **60 seconds**.

   10. Expand the **Advanced settings**,

* Under **Permissions**, let it be as default, i.e **Create or update IAM role**. New IAM role with required permission would be created and will be assigned to this Kinesis delivery stream.

   11. Click on **Create delivery stream**button.



## **Task 9: Creating and configuring Kinesis Agent**

Let us configure a Kinesis agent which will collect data and send it to Kinesis Data Streams.

1. SSH into the EC2 instance.
2. Let us install the latest version of Kinesis agent on the instance.

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

* + Type “**y**” if asked in the installation process.

1. After installing the Kinesis agent, let us update the json file available in the path **/etc/aws-kinesis/agent.json**.
2. Edit the **agent.json**,

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

* + **Remove** all the content and paste the below JSON content.
  + **Note 1**: Make sure you copy the “**awsAccessKeyId**” , "**awsSecretAccessKey**" from the lab page and paste it the JSON code wherever required.
  + **Note 2**: Make sure the “filePattern” consists of the log file path which is default in this case and “**kinesisStream**” consists of the created Kinesis Data Stream name.
  + Press “**ctrl** + **x**” to save. Press “**y**” to save the modified changes and press **Enter** (Follow the commands to save the file carefully).

|  |
| --- |
| {      "cloudwatch.emitMetrics": true,      "kinesis.endpoint": "",      "firehose.endpoint": "",      "awsAccessKeyId": "A\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*",      "awsSecretAccessKey": "B\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*",      "flows": [        {          "filePattern": "/var/log/httpd/access\_log",          "kinesisStream": "whiz-data-stream",          "partitionKeyOption": "RANDOM"        }      ]  } |

1. Make sure you specify the “**awsAccessKeyId**” "**awsSecretAccessKey**" from the lab page.
2. The name of the kinesis stream ( “**kinesisStream**” ) to which the agent sends data. Change the kinesisStream name according to the name you created.
3. Whenever you change the configuration file (**agent.json**), you must stop and start the agent, using the commands.

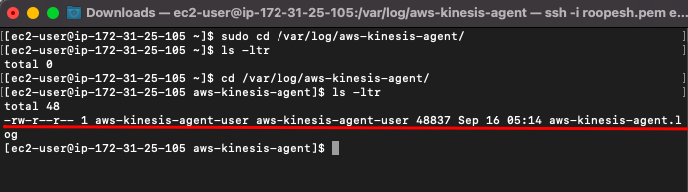
sudo service aws-kinesis-agent stop

sudo service aws-kinesis-agent start

1. Once the agent is started for the first time, a log file will be created. Check the log file using the commands,

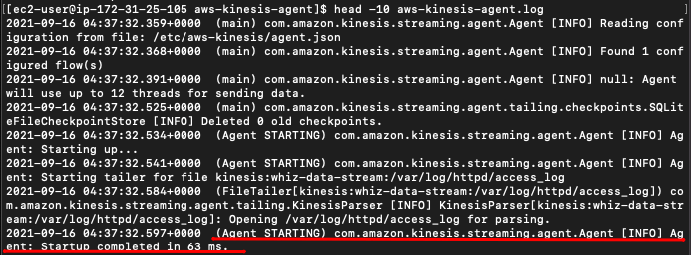
cd /var/log/aws-kinesis-agent/

ls -ltr



     9. You can check if the service is started properly by going through the log.

head -10 aws-kinesis-agent.log

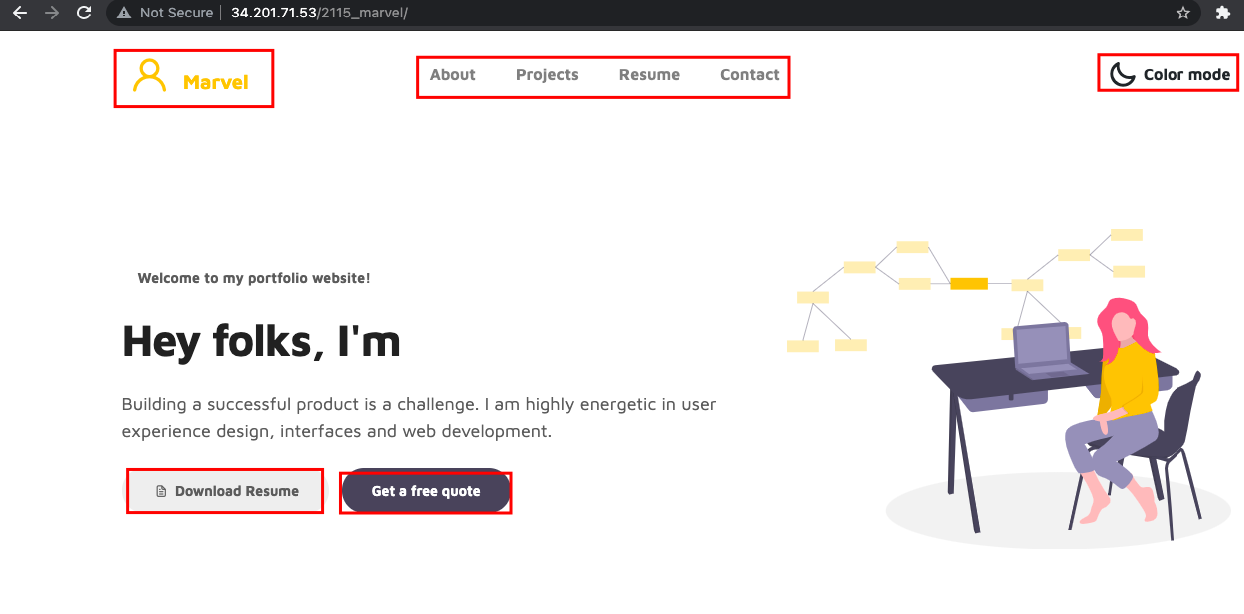


    10. We can see that the agent is successfully started.

## **Task 10: Testing the real-time streaming of data**

Let us test by hosting the above sample website on multiple browsers or do some click activity on the website. The related logs will be collected on the listed S3 bucket.

1. To test the data streaming, paste your **IP\_Address/folder\_name** in the multiple browsers and press enter.
   * **http://34.233.120.188/2115\_marvel/**
   * **Note:** The folder\_name is the unzipped folder of the sample website which we have noted earlier in the Task 5.
2. Once you have followed the above step, click on the website links present to create more logs.



* **Note:** We are clicking the links in the sample website website to generate logs which will be streamed to the created S3 Bucket.

1. Navigate to **S3** by clicking on the **Services** menu, under the **Storage** section.

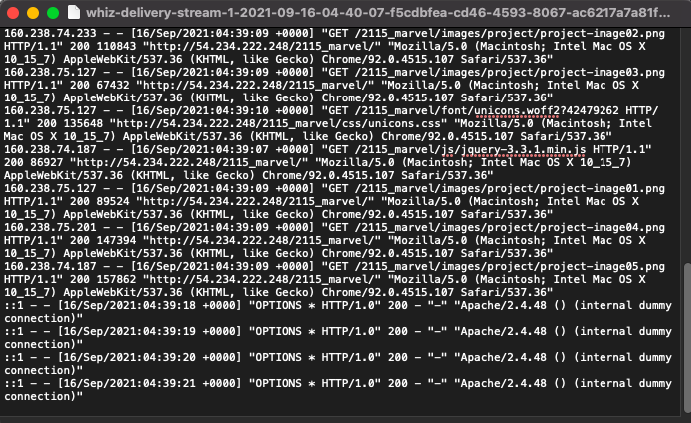
* **Note:** Wait for 3-5 minutes, if you are not able to see the logs.

1. You will see a hierarchy of folders with **year** > **month** > **date** > **hour**.
2. Click on the date or hour to see the logs created.

A screenshot of a computer

Description automatically generated with medium confidence

1. Click on the log and select **Open** and save the file.
2. Open the log file in any text editor in the local and check the logs.

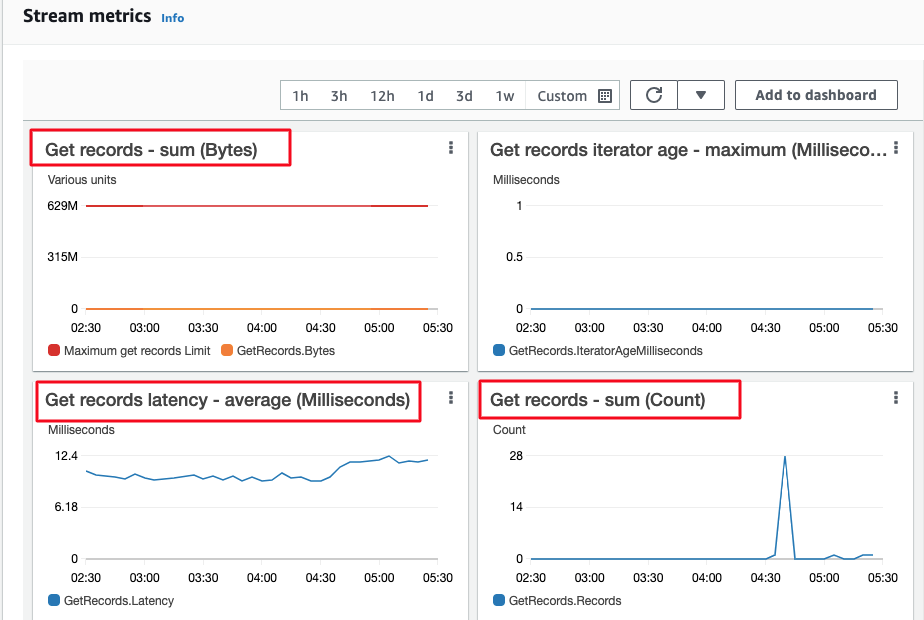


* **Note:** The more the clicks in the page, the more the logs are generated. In this demo webpage, we have only 1 page. So try to open the webpage in many browsers and click on the links to generate the logs.

## **Task 11: Checking the CloudWatch metrics of Kinesis Data Stream and Firehose**

Let us the check the CloudWatch metrics of Kinesis Data Stream which records the data and Kinesis Delivery stream which reads the data from Data Stream.

1. Make sure you are inthe **US East (N. Virginia) us-east-1** Region.
2. Navigate to **Kinesis** by clicking on the **Services** menu, under the **Analytics** section.
3. Click on the created data stream and navigate to the **Monitoring** tab. You will be able to see the graph according to the logs generated.



1. On the left navigation panel, click on the **Delivery streams**.
2. Click on the created delivery stream and navigate to the **Monitoring** tab. You will be able to see the graph.

A screenshot of a computer

Description automatically generated with medium confidence

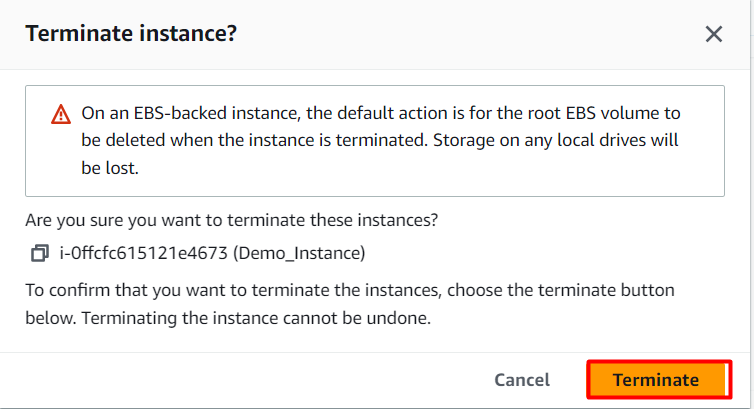
### Do you know ?

Amazon Kinesis Data Streams supports the concept of shards, which are the fundamental units of data capacity within a stream. Each shard in a Kinesis Data Stream can handle up to 1 MB/sec of data input and 2 MB/sec of data output. However, with the use of Kinesis Data Firehose, you can easily ingest data into Kinesis Data Streams at a much higher rate, exceeding the individual shard limits.

## **Task 12: Delete AWS Resources**

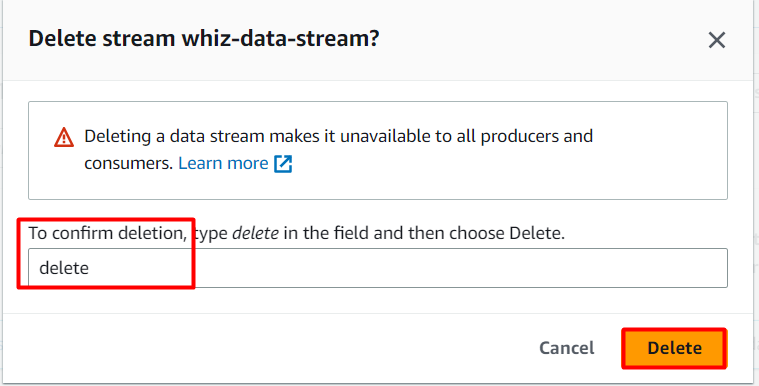
### Terminating EC2 Instance

1. Make sure you are in the **US East** (**N.Virginia) us-east-1** Region.
2. Navigate to the **Services** menu at the top left corner and click on **EC2** present under the **Compute** section.
3. Click on **Instances**from the left navigation menu.
4. Select the created instance and click on **Instance state**.
5. From the drop-down menu select **Terminate instance**.
6. Confirm the terminate by clicking on **Terminate**button.



### Deleting Kinesis Data Streams

1. Make sure you are in the **US East** (**N.Virginia) us-east-1** Region.
2. Navigate to **Kinesis** by clicking on the **Services** menu, under the **Analytics** section.
3. On the left panel, click on the **Data streams**.
4. Select the created data stream and click on the **Actions** button.
5. Select **Delete** from the drop-down.
6. Confirm by typing **Delete** and click on **Delete** button.



### Deleting Kinesis Delivery Streams

1. Make sure you are in the **US East** (**N.Virginia) us-east-1** Region.
2. Navigate to **Kinesis** by clicking on the **Services** menu, under the **Analytics** section.
3. On the left panel, click on the **Delivery streams**.
4. Select the created delivery stream.
5. Select **Delete**.
6. Confirm by typing the delivery stream name in the field and click on **Delete** button.

A screenshot of a computer

Description automatically generated with low confidence

# Completion and Conclusion

1. You have successfully launched an EC2 Instance.
2. You have successfully hosted a sample website.
3. You have successfully set file permissions to httpd.
4. You have successfully created Kinesis data stream.
5. You have successfully created a S3 Bucket.
6. You have successfully created Kinesis Data Firehose.
7. You have successfully created and configured Kinesis Agent.
8. You have successfully tested the real-time streaming of data.
9. You have successfully checked the CloudWatch metrics of Kinesis Data Streams and Data Firehose.

# End Lab