## **Prerequisites**

* If you have not already done so, sign up for an account in [Setting Up](https://docs.aws.amazon.com/athena/latest/ug/setting-up.html).
* Using the same AWS Region (for example, US West (Oregon)) and account that you are using for Athena, [Create a bucket in Amazon S3](https://docs.aws.amazon.com/AmazonS3/latest/user-guide/create-bucket.html) to hold your Athena query results.

## **Step 1: Create a Database**

You first need to create a database in Athena.

To create an Athena database

1. Open the Athena console at [https://console.aws.amazon.com/athena/](https://console.aws.amazon.com/athena/home).
2. If this is your first time to visit the Athena console in your current AWS Region, choose Explore the query editor to open the query editor. Otherwise, Athena opens in the query editor.
3. Choose View Settings to set up a query result location in Amazon S3.  
   
                           Choose View Settings.
                       
4. On the Settings tab, choose Manage.  
   
                           Choose Manage.
                       
5. For Manage settings, do one of the following:
   * In the Location of query result box, enter the path to the bucket that you created in Amazon S3 for your query results. Prefix the path with s3://.
   * Choose Browse S3, choose the Amazon S3 bucket that you created for your current Region, and then choose Choose.
6. 
                           Specify a location in Amazon S3 to receive query results from
                               Athena.
                       
7. Choose Save.
8. Choose Editor to switch to the query editor.  
   
                           Choose Editor.
                       
9. On the right of the navigation pane, you can use the Athena query editor to enter and run queries and statements.  
   
                           The query editor in the Athena console.
                       
10. To create a database named mydatabase, enter the following CREATE DATABASE statement.
11. CREATE DATABASE mydatabase
12. Choose Run or press **Ctrl+ENTER**.
13. From the Database list on the left, choose mydatabase to make it your current database.  
    
                            Choose the database that you created.
                        

## **Step 2: Create a Table**

Now that you have a database, you can create an Athena table for it. The table that you create will be based on sample Amazon CloudFront log data in the location s3://athena-examples-*myregion*/cloudfront/plaintext/, where *myregion* is your current AWS Region.

The sample log data is in tab-separated values (TSV) format, which means that a tab character is used as a delimiter to separate the fields. The data looks like the following example. For readability, the tabs in the excerpt have been converted to spaces and the final field shortened.

2014-07-05 20:00:09 DFW3 4260 10.0.0.15 GET eabcd12345678.cloudfront.net /test-image-1.jpeg 200 - Mozilla/5.0[...]

2014-07-05 20:00:09 DFW3 4252 10.0.0.15 GET eabcd12345678.cloudfront.net /test-image-2.jpeg 200 - Mozilla/5.0[...]

2014-07-05 20:00:10 AMS1 4261 10.0.0.15 GET eabcd12345678.cloudfront.net /test-image-3.jpeg 200 - Mozilla/5.0[...]

To enable Athena to read this data, you could run a CREATE EXTERNAL TABLE statement like the following. The statement that creates the table defines columns that map to the data, specifies how the data is delimited, and specifies the Amazon S3 location that contains the sample data.

**Note**

For the LOCATION clause, specify an Amazon S3 folder location, not a specific file. Athena scans all of the files in the folder that you specify.

CREATE EXTERNAL TABLE IF NOT EXISTS cloudfront\_logs (

`Date` DATE,

Time STRING,

Location STRING,

Bytes INT,

RequestIP STRING,

Method STRING,

Host STRING,

Uri STRING,

Status INT,

Referrer STRING,

ClientInfo STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

LINES TERMINATED BY '\n'

LOCATION 's3://athena-examples-*my-region*/cloudfront/plaintext/';

The example creates a table called cloudfront\_logs and specifies a name and data type for each field. These fields become the columns in the table. Because date is a [reserved word](https://docs.aws.amazon.com/athena/latest/ug/reserved-words.html#list-of-ddl-reserved-words), it is escaped with backtick (`) characters. ROW FORMAT DELIMITED means that Athena will use a default library called [LazySimpleSerDe](https://docs.aws.amazon.com/athena/latest/ug/lazy-simple-serde.html) to do the actual work of parsing the data. The example also specifies that the fields are tab separated (FIELDS TERMINATED BY '\t') and that each record in the file ends in a newline character (LINES TERMINATED BY '\n). Finally, the LOCATION clause specifies the path in Amazon S3 where the actual data to be read is located. If you have your own tab or comma-separated data, you can use a CREATE TABLE statement like this.

Returning to the sample data, here is a full example of the final field ClientInfo:

Mozilla/5.0%20(Android;%20U;%20Windows%20NT%205.1;%20en-US;%20rv:1.9.0.9)%20Gecko/2009040821%20IE/3.0.9

As you can see, this field is multivalued. If the CREATE TABLE statement specifies tabs as field delimiters, the separate components inside this particular field can't be broken out into separate columns. To create columns from the values inside the field, you can use a regular expression (regex) that contains regex groups. The regex groups that you specify become separate table columns. To use a regex in your CREATE TABLE statement, use syntax like the following. This syntax instructs Athena to use the [Regex SerDe](https://docs.aws.amazon.com/athena/latest/ug/regex-serde.html) library and the regular expression that you specify.

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'

WITH SERDEPROPERTIES ("input.regex" = "*regular\_expression*")

Regular expressions can be useful for creating tables from complex CSV or TSV data but can be difficult to write and maintain. Fortunately, there are other libraries that you can use for formats like JSON, Parquet, and ORC. For more information, see [Supported SerDes and Data Formats](https://docs.aws.amazon.com/athena/latest/ug/supported-serdes.html).

Now you are ready to create the table in the Athena query editor. The CREATE TABLE statement and regex are provided for you.

To create a table in Athena

1. In the navigation pane, for Database, make sure that mydatabase is selected.
2. To give yourself more room in the query editor, you can choose the arrow icon to collapse the navigation pane.  
   
                           Choose the arrow to collapse the navigation pane.
                       
3. To create a tab for a new query, choose the plus (+) sign in the query editor. You can have up to ten query tabs open at once.  
   
                           Choose the plus icon to create a new query.
                       
4. To close one or more query tabs, choose the arrow next to the plus sign. To close all tabs at once, choose the arrow, and then choose Close all tabs.  
   
                           Choose the arrow icon to close one or more query tabs.
                       
5. In the query pane, enter the following CREATE EXTERNAL TABLE statement. The regex breaks out the operating system, browser, and browser version information from the ClientInfo field in the log data.

CREATE EXTERNAL TABLE IF NOT EXISTS cloudfront\_logs (

`Date` DATE,

Time STRING,

Location STRING,

Bytes INT,

RequestIP STRING,

Method STRING,

Host STRING,

Uri STRING,

Status INT,

Referrer STRING,

os STRING,

Browser STRING,

BrowserVersion STRING

)

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'

WITH SERDEPROPERTIES (

"input.regex" = "^(?!#)([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+[^\(]+[\(]([^\;]+).\*\%20([^\/]+)[\/](.\*)$"

1. ) LOCATION 's3://athena-examples-*myregion*/cloudfront/plaintext/';
2. In the LOCATION statement, replace *myregion* with the AWS Region that you are currently using (for example, us-west-1).
3. Choose Run.  
   The table cloudfront\_logs is created and appears under the list of Tables for the mydatabase database.

## **Step 3: Query Data**

Now that you have the cloudfront\_logs table created in Athena based on the data in Amazon S3, you can run SQL queries on the table and see the results in Athena. For more information about using SQL in Athena, see [SQL Reference for Amazon Athena](https://docs.aws.amazon.com/athena/latest/ug/ddl-sql-reference.html).

To run a query

1. Choose the plus (+) sign to open a new query tab and enter the following SQL statement in the query pane.

SELECT os, COUNT(\*) count

FROM cloudfront\_logs

WHERE date BETWEEN date '2014-07-05' AND date '2014-08-05'

1. GROUP BY os
2. Choose Run.  
   The results look like the following:  
   
                           Viewing query results in the Athena console.
                       
3. To save the results of the query to a .csv file, choose Download results.  
   
                           Downloading query results in CSV format.
                       
4. To view or run previous queries, choose the Recent queries tab.  
   
                           Choose Recent queries to view previous
                               queries.
                       
5. To download the results of a previous query from the Recent queries tab, select the query, and then choose Download results. Queries are retained for 45 days.  
   
                           Viewing and downloading recent queries in the Athena
                               console.
                         
   For more information, see [Working with Query Results, Recent Queries, and Output Files](https://docs.aws.amazon.com/athena/latest/ug/querying.html).