

**Real-Time Reddit Streaming Solution**

**Self-Guided Tutorial**

*Kinesis Firehose, S3, Glue, Athena Use-case*

**Updated July 2019**

Table of Contents

[2. Overview 3](#_Toc20993837)

[3. What you will accomplish 3](#_Toc20993838)

[4. Prerequisites 3](#_Toc20993839)

[5. Create your Reddit bot account 4](#_Toc20993840)

[6. Set up an S3 Bucket 5](#_Toc20993841)

[7. Deploy the AWS Glue data catalog in CloudFormation 6](#_Toc20993842)

[8. Set up Kinesis Firehose Delivery Stream 8](#_Toc20993843)

[9. Create a Key Pair for your streaming server 10](#_Toc20993844)

[10. Deploy the EC2 streaming server in CloudFormation 11](#_Toc20993845)

[11. Monitor the delivery stream 13](#_Toc20993846)

[13. Use Athena to develop insights 14](#_Toc20993847)

[13. Clean up the environment 16](#_Toc20993848)

[14. Conclusion 19](#_Toc20993849)

[Appendix 20](#_Toc20993850)

[I. Troubleshooting your streaming application 20](#_Toc20993851)

# 2. Overview

AWS provides several key services for an easy way to quickly deploy and manage data streaming in the cloud. Reddit is a popular social news aggregation, web content rating, and discussion website. At peak times, Reddit can see over 300,000 comments and 35,000 submissions an hour. The Reddit API offers developers a simple way to collect all of this data, which is a perfect use case to learn how to use Kinesis Firehose, S3, Glue, and Athena.

In this tutorial, you will play the role of a data architect looking to modernize a company’s streaming pipeline. You will create a Kinesis Firehose delivery stream from an EC2 server to an S3 data lake. With the help of AWS Glue and Amazon Athena, you’ll be able to develop insights on the data as it accumulates in your data lake.

# 3. What you will accomplish

* Create a Reddit App using the Reddit developer site
* Provision an S3 bucket to act as a data lake and the target for your stream data
* Provision a Kinesis Firehose Delivery Stream that will accept data from various sources and deliver it to the S3 bucket
* Deploy and run an EC2 Streaming python app via CloudFormation
* Create a Glue data catalog via CloudFormation to provide schemas and structure to your data
* Use Athena to directly query your S3 bucket with SQL

# 4. Prerequisites

This tutorial requires:

* A laptop with Wi-Fi running Microsoft Windows, Mac OS X, or Linux
* An Internet browser of Chrome or Firefox
* An AWS account to provision the AWS infrastructure
* **Skill level:** A basic understanding of desktop computing is helpful but not required
* **AWS experience:** Prior knowledge of base AWS infrastructure (VPC, EC2, S3) is helpful, but not required to complete this exercise

# 5. Create your Reddit bot account

1. Go to: <https://www.reddit.com/register/>
2. Follow prompts to create new reddit account:
   1. Provide email address
   2. Choose username and password
   3. Click Finish
3. Once your account is created, go to: <https://www.reddit.com/prefs/apps/>
4. Select **“are you a developer? Create an app...”**
5. Give it a name.
6. Select **script**. 🡨 This is important!
7. For **about url** and **redirect uri**, use <http://127.0.0.1>
8. You will now get a **client\_id** (underneath web app) and secret
9. Keep track of your Reddit account **username**, **password**, app **client\_id** (in blue box), and app **secret** (in red box). These will be used in tutorial Step 11



*Further Learning / References:*

<https://praw.readthedocs.io/en/latest/getting_started/quick_start.html>

*Next steps:* Our app information is registered now. Before you begin setting up the server or the delivery stream, you need a place to store the data that will be generated.

# 6. Set up an S3 Bucket

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>
2. Choose **Create bucket**
3. In the Bucket name field, type a unique DNS-compliant name for your new bucket. (The example screen shot uses the bucket name admin-created. You cannot use this name because S3 bucket names must be unique.) Create your own bucket name using the following naming guidelines:
   * + The name must be unique across all existing bucket names in Amazon S3
     + Example: **reddit-analytics-bucket-<random number here>**
     + After you create the bucket you cannot change the name, so choose wisely
     + Choose a bucket name that reflects the objects in the bucket because the bucket name is visible in the URL that points to the objects that you're going to put in your bucket
     + For information about naming buckets, see Rules for Bucket Naming in the Amazon Simple Storage Service Developer Guide
4. For Region, choose **US East (N. Virginia)** as the region where you want the bucket to reside
5. Keep defaults and continue clicking **Next**
6. Choose **Create**

Now that you’ve created a bucket, let’s set up a delivery stream for your data

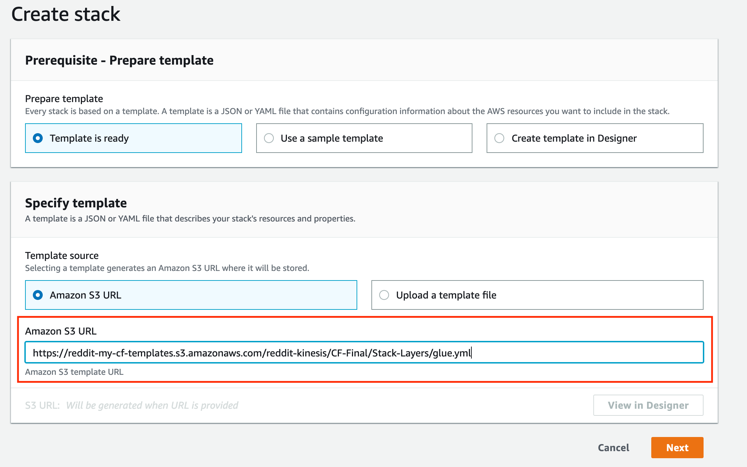
*Next steps:* S3 is a place to store many different kinds of data / files. To provide the data files with structure that services can reference, you need to set up a data catalog. AWS Glue is the perfect service for this use case.

# 7. Deploy the AWS Glue data catalog in CloudFormation

In this step we will be using a tool called CloudFormation. Instead of going through the AWS console and creating glue databases and glue tables click by click, we can utilize CloudFormation to deploy the infrastructure quickly and easily

1. Open the AWS CloudFormation console at <https://console.aws.amazon.com/cloudformation>
2. If this is a new AWS CloudFormation account, click Create New Stack Otherwise, click Create Stack
3. In the Template section, select Specify an Amazon S3 Template URL to type or paste the URL for the sample WordPress template, and then click Next:

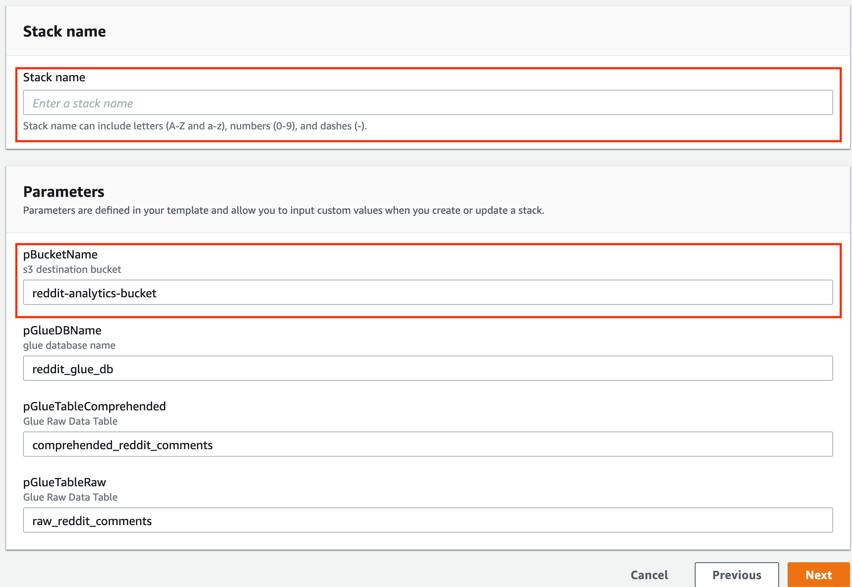
<https://s3-us-east-1.amazonaws.com/cloudformation-templates-us-east-1/Streaming_App_Glue_Data_Catalog.template>



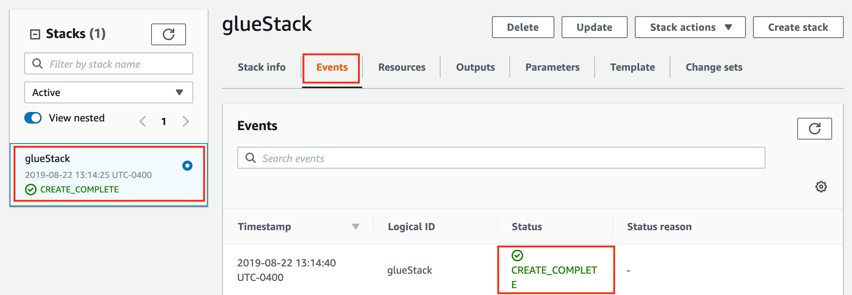
*Note:*

*AWS CloudFormation templates that are stored in an S3 bucket must be accessible to the user who is creating the stack, and must be located in the same region as the stack that is being created. Therefore, if the S3 bucket is located in the us-east-1 Region, the stack must also be created in us-east-1*

1. Decide on your stack name
2. Under **pBucketName** set your bucket name from the previous step



1. Continue until the last step and click Create stack
2. Click on **Events** tab. Wait until the stack status is **CREATE\_COMPLETE**



*Further Learning / References:*

<https://docs.aws.amazon.com/glue/latest/dg/what-is-glue.html>

<https://docs.aws.amazon.com/glue/latest/dg/populate-with-cloudformation-templates.html>

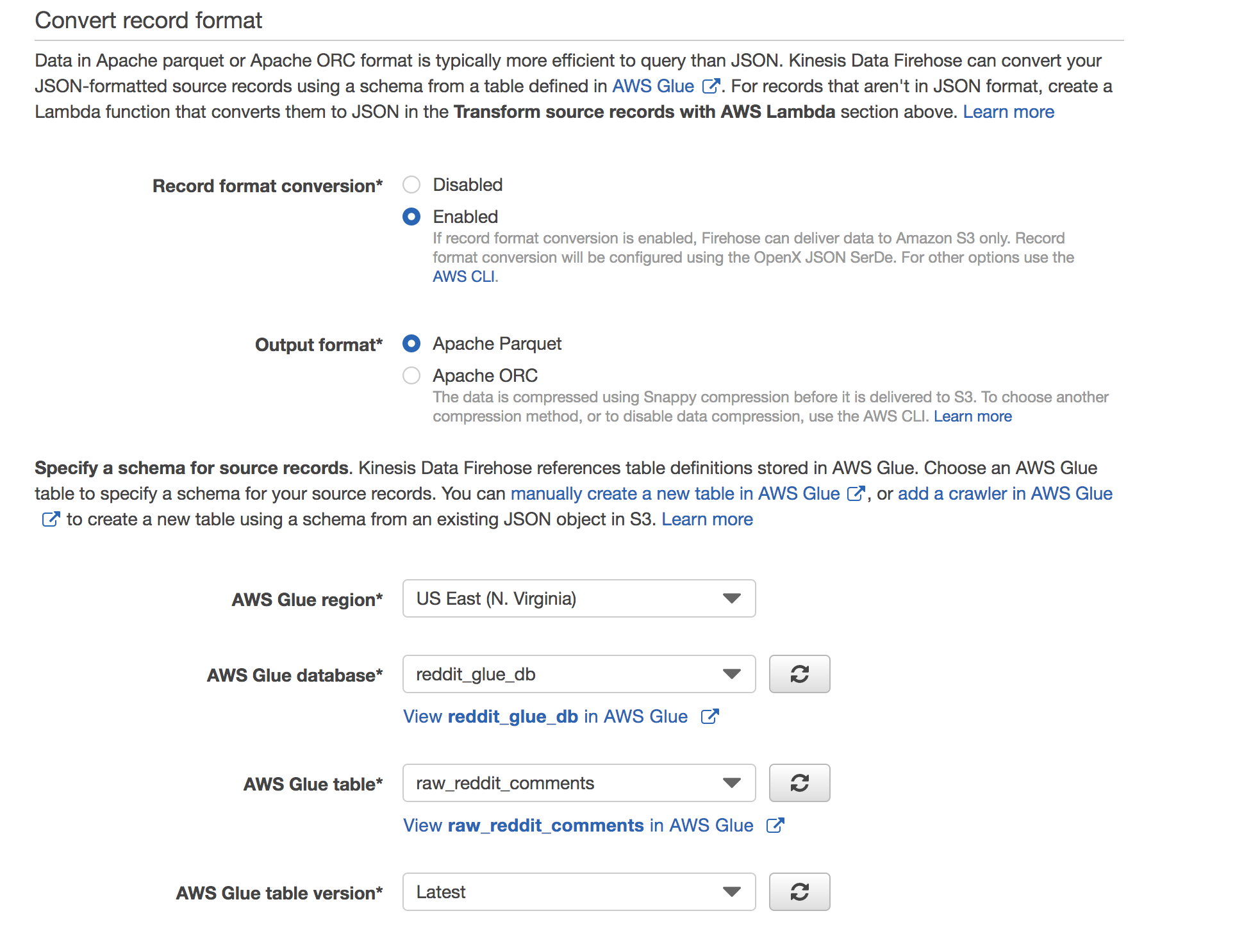
*Next steps:* Now you have a destination for your data (S3) and a formatted structure (AWS Glue). Next, let’s deploy the pipes that will allow data to travel between services.

# 8. Set up Kinesis Firehose Delivery Stream

1. Open the Kinesis Data Firehose console at <https://console.aws.amazon.com/firehose/> or select Kinesis in the **Services** dropdown
2. Choose **Create Delivery Stream**
3. Delivery stream name – Type a name for the delivery stream

Example: **raw-reddit-comment-delivery-stream**

1. Keep default settings on Step 1 - you will be using a direct PUT as source. Scroll down and click **Next**
2. In Step 2, **enable record format conversion** by using the following settings:



1. Click **Next**
2. On the Destination page, choose the following options

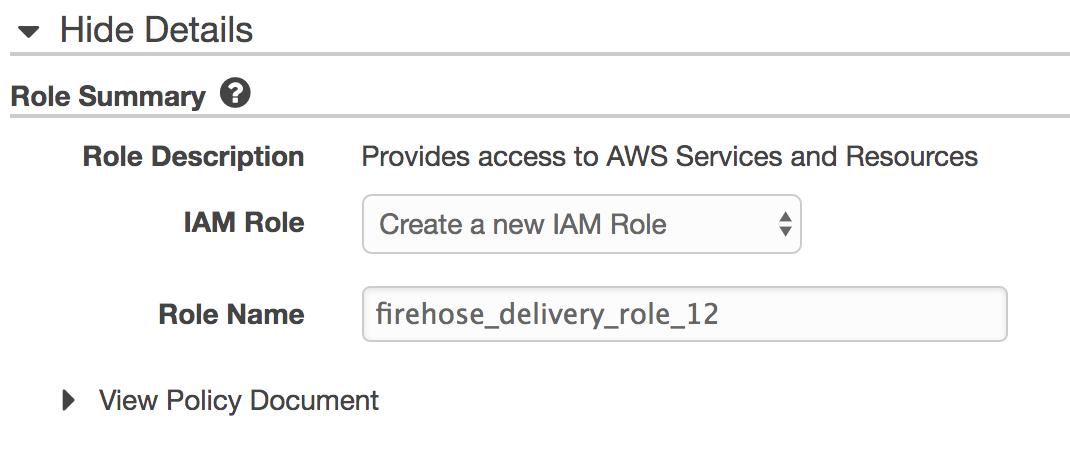
**Destination** – Choose Amazon S3

**S3 bucket** – Choose an existing bucket created in tutorial Step 6

**S3 prefix** – add "**raw\_reddit\_comments**" as prefix

**S3 error prefix** - add "**raw\_reddit\_comments\_error**" as prefix

1. Choose **Next**
2. On the Configuration page, **Change Buffer time to 60 seconds**
3. For IAM Role, click **Create new or choose**
4. For the IAM Role summary, use the following settings:



1. Choose **Allow**
2. You should return to the Kinesis Data Firehose delivery stream set-up steps in the Kinesis Data Firehose console
3. Choose **Next**
4. On the Review page, review your settings, and then choose **Create Delivery Stream**

*Further Learning / References:*

<https://docs.aws.amazon.com/firehose/latest/dev/what-is-this-service.html>

<https://docs.aws.amazon.com/firehose/latest/dev/basic-deliver.html>

*Next steps:* The pipeline and destination are now available for use. In the next several steps, you will be creating the python application that generates Reddit comment data.

# 9. Create a Key Pair for your streaming server

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/> or select EC2 under **Services** dropdown
2. In the navigation pane, under NETWORK & SECURITY, choose **Key Pairs**

Note: The navigation pane is on the left side of the Amazon EC2 console. If you do not see the pane, it might be minimized; choose the arrow to expand the pane

1. Choose **Create Key Pair**
2. For Key pair name, enter a name for the new key pair (ex: RedditBotKey), and then choose **Create**
3. The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is .pem. **Save the private key file in a safe place**

*Important: This is the only chance for you to save the private key file. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance*

*Further Learning / References:*

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html>

*Next steps:* A key pair will allow you to securely access a server. In the next steps, you will deploy the server and connect to the server using the key pair.

# 10. Deploy the EC2 streaming server in CloudFormation

In this step you will be using a tool called CloudFormation. Instead of going through the AWS console and creating an EC2 instance click by click, you can utilize CloudFormation to deploy the infrastructure quickly. This CloudFormation template has EC2 user data to set up the machine. The EC2 user data achieves the following:

* + Installs python 3.6 and several libraries needed for the script to run
  + Clones a GitHub repository that contains the python script
  + Updates python script with custom permissions and parameters
  + Executes the script to begin the data stream

1. Open the AWS CloudFormation console at <https://console.aws.amazon.com/cloudformation> or select CloudFormation under the **Services** dropdown
2. Click **Create New Stack** / **Create Stack**
3. In the Template section, select Specify an Amazon S3 Template URL to type or paste the URL for the sample WordPress template

<https://s3-us-east-1.amazonaws.com/cloudformation-templates-us-east-1/Streaming_App_EC2_Server.template>

1. Click **Next**
2. Provide a stack name (ex: **reddit-stream-server**)
3. For **pKeyName** and provide the key name that you created in tutorial Step 9
4. Use your reddit app info and reddit account for the parameters **pRedditAppSecret**, **pRedditClientID**, **pRedditUsername**, and **pRedditPassword**
5. You can choose to leave the rest of the parameters as their default values.
6. Continue to click **Next**
7. On the last step, acknowledge IAM resource creation and click **Create Stack**
8. Wait for your EC2 instance to be created.
9. Make a note of the **Public IP** and **Public DNS Name** given to the newly created instance. You can find these in the Cloudformation Outputs tab.
10. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/> or select EC2 under **Services** dropdown
11. Select **INSTANCES** in the navigation pane
12. Ensure that an EC2 instance has been created and running. (This can take several minutes to deploy)

*Further Learning / References:*

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html>

*Next steps:* Now that the EC2 instance is provisioned and the script is running, the data is streaming to Kinesis Firehose. In the next step we’ll monitor the data as it moves through the delivery stream and into S3.

# 11. Monitor the delivery stream

1. Open the Amazon Kinesis Firehose console at <https://console.aws.amazon.com/firehose/> or select Kinesis in the **Services** dropdown
2. Select the delivery stream created in step 8.
3. Select **Monitoring** Tab
4. Click refresh button over the next 3 minutes. You should start to see records coming in
5. If you are still not seeing data after 3-5 minutes, go to **Appendix I** for troubleshooting.
6. Now let’s check the S3 bucket
7. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/> or select S3 in the **Services** dropdown
8. Click the bucket name of the bucket that you created in step 2
9. Verify that records are being PUT into your s3 bucket

Now that data is streaming into s3, let’s build a data catalog so that you can query our s3 files

*Further Learning / References:*

<https://docs.aws.amazon.com/firehose/latest/dev/monitoring.html>

*Next steps:* Now that you have all of our infrastructure in place, you can finally begin to analyze the data currently streaming into our data lake. You will use Amazon Athena, a great tool for ad-hoc queries on S3 data.

# 13. Use Athena to develop insights

1. Open the AWS CloudFormation console at <https://console.aws.amazon.com/athena/> or select Athena in the **Services** dropdown
2. Choose the glue database (**reddit\_glue\_db**) populated on the left view
3. Select the table (**raw\_reddit\_comments**) to view the table schema
4. You should now be able to use SQL to query the table (S3 data)

Here are some example queries to begin exploring the data streaming into S3:

|  |
| --- |
|  |
|  | -- total number of comments | |
|  | select count(\*) | |
|  | from raw\_reddit\_comments; | |
|  |  | |
|  | -- general sentiment of reddit Today | |
|  |  | |
|  | select round(avg(comment\_tb\_sentiment), 4) as avg\_comment\_tb\_sentiment | |
|  | from raw\_reddit\_comments | |
|  | where comment\_date | |
|  | like '%2019-08-22%'; | |
|  |  | |
|  | -- total comments collected per subreddits | |
|  |  | |
|  | select count(\*) as num\_comments, subreddit | |
|  | from raw\_reddit\_comments | |
|  | group by subreddit  order by num\_comments DESC; | |
|  |  | |
|  | -- average sentiment per subreddits | |
|  |  | |
|  | select round(avg(comment\_tb\_sentiment), 4) as avg\_comment\_tb\_sentiment, subreddit | |
|  | from raw\_reddit\_comments | |
|  | group by subreddit  order by avg\_comment\_tb\_sentiment DESC; | |
|  |  | |
|  | -- list all Subreddits | |
|  |  | |
|  | select distinct(subreddit) | |
|  | from raw\_reddit\_comments; | |
|  |  | |
|  |  | |
|  | -- top 10 most positive comments by subreddit | |
|  |  | |
|  |
|  |
|  | select subreddit, comment\_body | |
|  | from raw\_reddit\_comments | |
|  | where subreddit = '${subreddit}' | |
|  | order by comment\_tb\_sentiment DESC  limit 10; | |
|  |
|  | |
|  | -- most active subreddits and their sentiment | |
|  |  | |
|  | select subreddit, count(\*) as num\_comments, round(avg(comment\_tb\_sentiment), 4) as avg\_comment\_tb\_sentiment | |
|  | from raw\_reddit\_comments | |
|  | group by subreddit | |
|  | order by num\_comments DESC; | |
|  |  | |
|  | -- search term frequency by subreddit where comments greater than 5 | |
|  |  | |
|  | select subreddit, count(\*) as comment\_occurrences | |
|  | from raw\_reddit\_comments | |
|  | where LOWER(comment\_body) like '%puppy%' | |
|  | group by subreddit | |
|  | having count(\*) > 5 | |
|  | order by comment\_occurrences desc; | |
|  |  | |
|  |
|  |
|  |
|  |
|  |
|  | -- search term sentiment by subreddit | |
|  |  | |
|  | select subreddit, round(avg(comment\_tb\_sentiment), 4) as avg\_comment\_tb\_sentiment | |
|  | from raw\_reddit\_comments | |
|  | where LOWER(comment\_body) like '%puppy%' | |
|  | group by subreddit | |
|  | having count(\*) > 5 | |
|  | order by avg\_comment\_tb\_sentiment desc; | |
|  | -- top 25 most positive comments about a search term | |
|  | select subreddit, author\_name, comment\_body, comment\_tb\_sentiment | |
|  | from raw\_reddit\_comments | |
|  | where LOWER(comment\_body) like '%puppy%' | |
|  | order by comment\_tb\_sentiment desc | |
|  | limit 25; | |
|  |  | |
|  |  | |
|  | -- total sentiment for search term | |
|  |  | |
|  | SELECT round(avg(comment\_tb\_sentiment), 4) as avg\_comment\_tb\_sentiment | |
|  | FROM ( | |
|  | SELECT subreddit, author\_name, comment\_body, comment\_tb\_sentiment | |
|  | FROM raw\_reddit\_comments | |
|  | WHERE LOWER(comment\_body) LIKE '%puppy%') | |
|  |  | |

*Further Learning / References:*

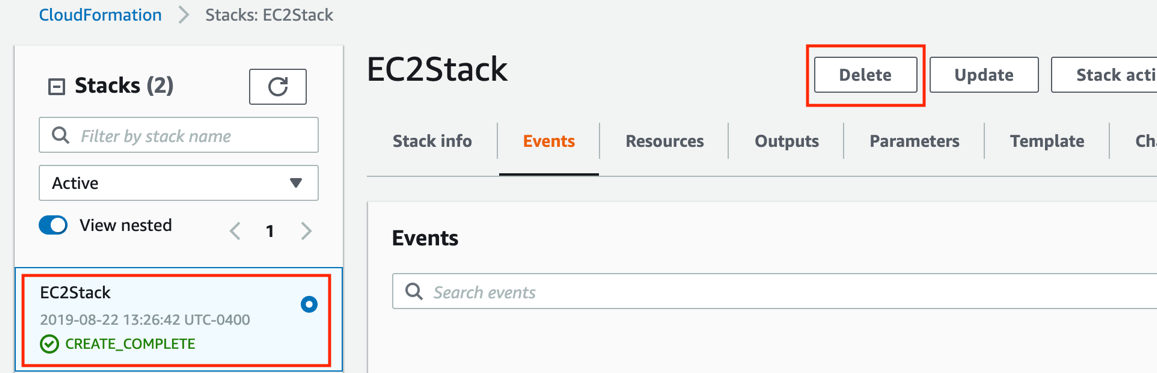
<https://docs.aws.amazon.com/athena/latest/ug/glue-athena.html>

*Next Steps:* Hopefully by now you have found some interesting insights into Reddit and the overall public sentiment. Athena is a great service for ad-hoc queries like this. You are approaching the end of this tutorial, so you will start terminating services and instances to prevent further billing.

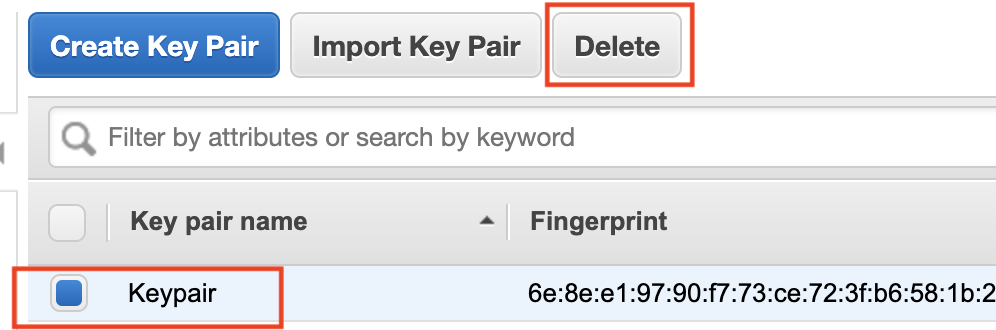
# 13. Clean up the environment

* **Do not skip this step! Leaving AWS resources without tearing down can result a bill in the end of the month. Make sure you follow the steps to remove the resources you’ve created**

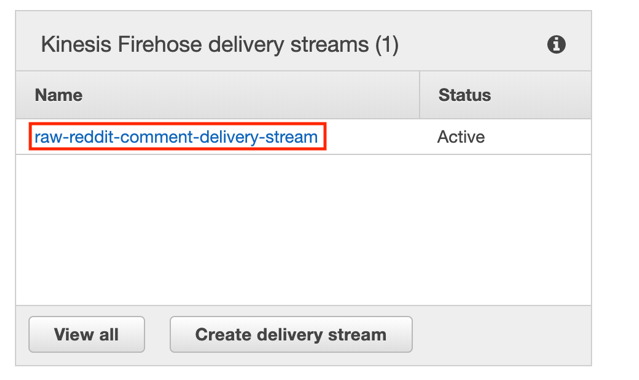
1. **EC2** – Our EC2 instance was created from a CloudFormation template, we’ll delete the stack and the key pair
   1. Open the AWS CloudFormation console at <https://console.aws.amazon.com/cloudformation> or select CloudFormation under **Services** dropdown
   2. On the left panel that says stacks – click on the EC2 stack you’ve created
   3. Click **Delete** on top of the pane



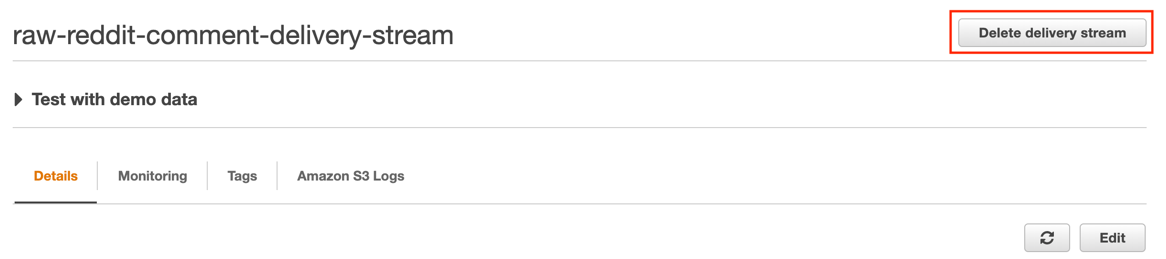
* 1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/> or select EC2 under **Services** dropdown
  2. In the navigation pane, under NETWORK & SECURITY, choose **Key Pairs**
  3. Click on the key pair name you’ve created and hit **Delete** on top



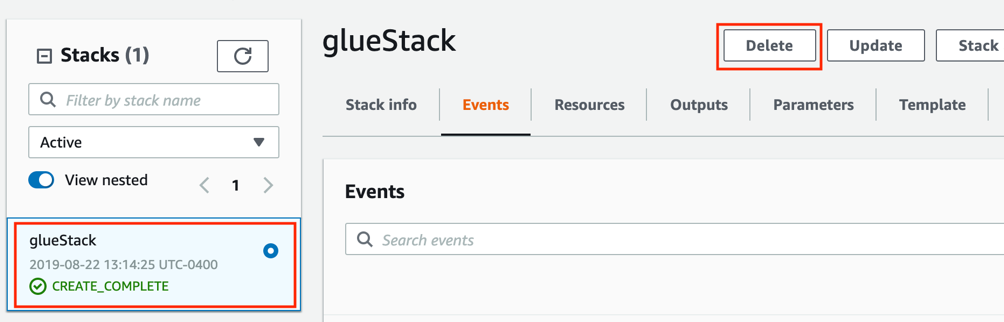
1. **Kinesis** –
   1. Open the Kinesis Data Firehose console at <https://console.aws.amazon.com/firehose/> or select Kinesis in the **Services** dropdown
   2. Click on the stream you’ve created at the top right corner (blue link).



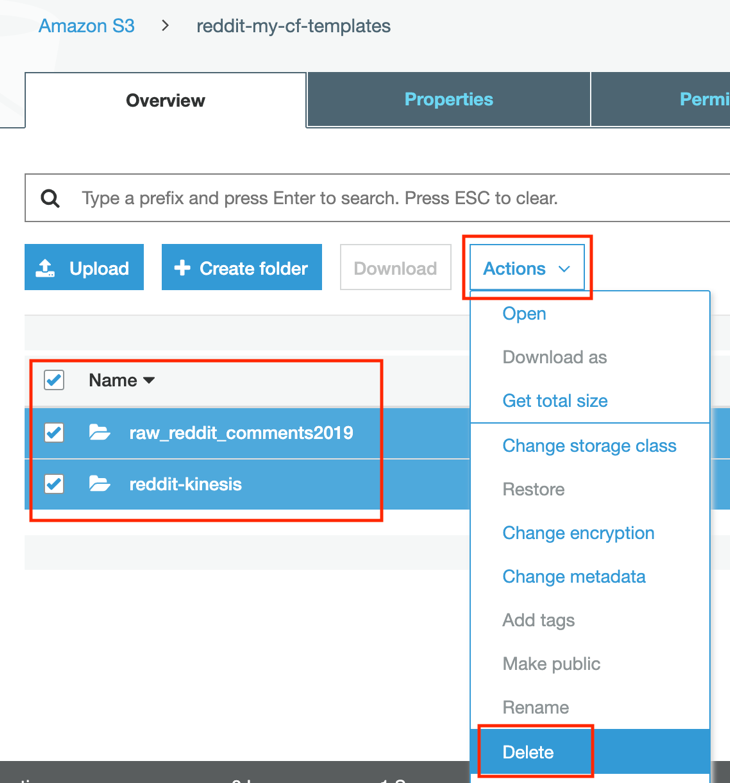
* 1. On the top right corner click on **Delete delivery stream**



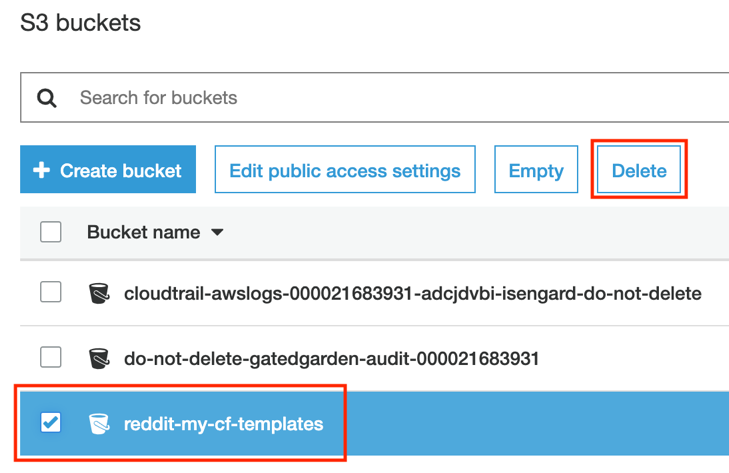
1. **Glue** –
   1. Open the AWS CloudFormation console at <https://console.aws.amazon.com/cloudformation>. or select Glue under **Services** dropdown
   2. On the left panel that says stacks – click on the EC2 stack you’ve created
   3. Click **Delete** on top of the pane



1. **S3** –
   1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>
   2. Click on the bucket you just created
   3. Mark all of the files and libraries in the bucket and hit **Actions -> Delete**



* 1. Now go to the main S3 console at <https://console.aws.amazon.com/s3/>.
  2. Mark the created bucket and hit **Delete**



# 14. Conclusion

In this tutorial, you have walked through the process of deploying a sample Python application that uses the Reddit API and AWS SDK for Python to stream Reddit data into Amazon Kinesis Firehose. You learned basic operations to deploy a real-time data streaming pipeline and data lake. Finally, you developed insights on the data using Amazon Athena’s ad-hoc SQL querying.

# Appendix

# I. Troubleshooting your streaming application

1. Find the Public IP address that you noted down in **Step 10** and the key pair you downloaded in **Step 9.**
2. Open up a Terminal
3. Go to the directory that your key pair was downloaded to.
4. SSH into the machine with the following command:

ssh -i <insert your key pair name here> ec2-user@<insert public IP address here>

1. Confirm that the correct credentials have been added to your application with the following command:

sudo cat /reddit/reddit-kinesis/praw.ini

1. Confirm that the correct delivery stream name was added to your application with the following command. Look for DeliveryStreamName=’<your delivery stream name>’

sudo cat /reddit/reddit-kinesis/comment-stream.py

1. If there are errors found, delete the CloudFormation stack that didn’t work properly. Go back and retry **Step 10.** If there are no errors you can check the logs:

sudo tail /tmp/reddit-stream.log

Some common errors include:

DEBUG:prawcore:Response: 503 (Reddit servers are down)

DEBUG:prawcore:Response: 403 (Your Reddit username/password is incorrect)