**March 2018**



**Intelligent Data Lake Workshop**

*Lab 2 – Glue, Athena, EMR, Redshift Spectrum and QuickSight*

Table of Contents

[Overview 3](#_Toc508543270)

[Populate the AWS Glue Data Catalog 4](#_Toc508543271)

[Transform Data with AWS Glue 11](#_Toc508543272)

[Query Data with Amazon Athena and AWS Glue Data Catalog 16](#_Toc508543273)

[Visualize Data with Amazon QuickSight and Amazon Athena 17](#_Toc508543274)

[Query Data with Amazon EMR and AWS Glue Data Catalog (Optional) 21](#_Toc508543275)

[Query Data with Amazon Redshift Spectrum and AWS Glue Data Catalog (Optional) 25](#_Toc508543276)

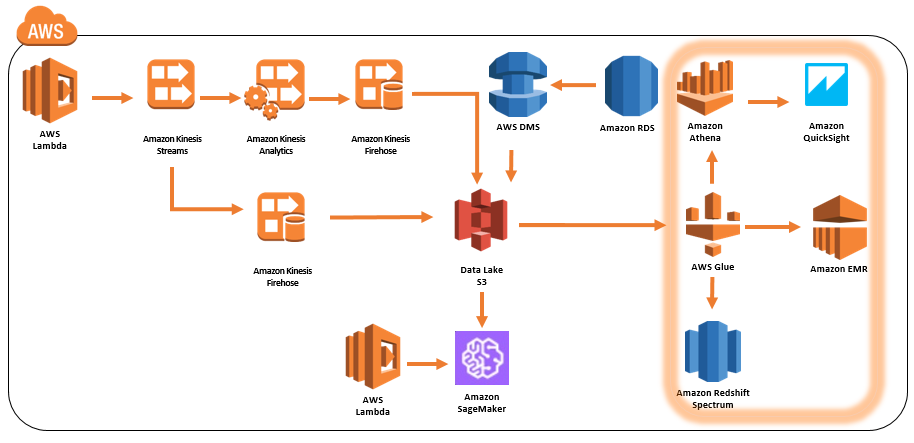
[Conclusion 31](#_Toc508543277)

[Appendix 32](#_Toc508543278)

# Overview

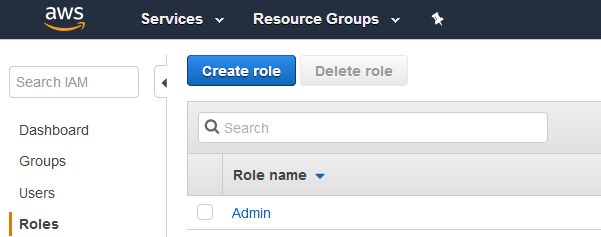
In lab 1, you used Kinesis Streams to collect and store the streaming IoT sensor data, then used Kinesis Analytics to process and analyze the streaming data continuously. You also used Amazon Kinesis Firehose to export both the raw and processed data into S3 for further analysis. In this Lab, you will use Glue Data Catalog to define schemas on the data and share it with disparate systems/services such as Amazon Athena, Amazon EMR and Amazon Redshift Spectrum. You will also explore how to use Glue ETL tools to transform raw data to Parquet format. Finally, you will use QuickSight to visualize the data stored on S3 with Athena. Diagram below depicts the complete architecture.

Diagram below with highlighted area depicts what you will be building in this lab.

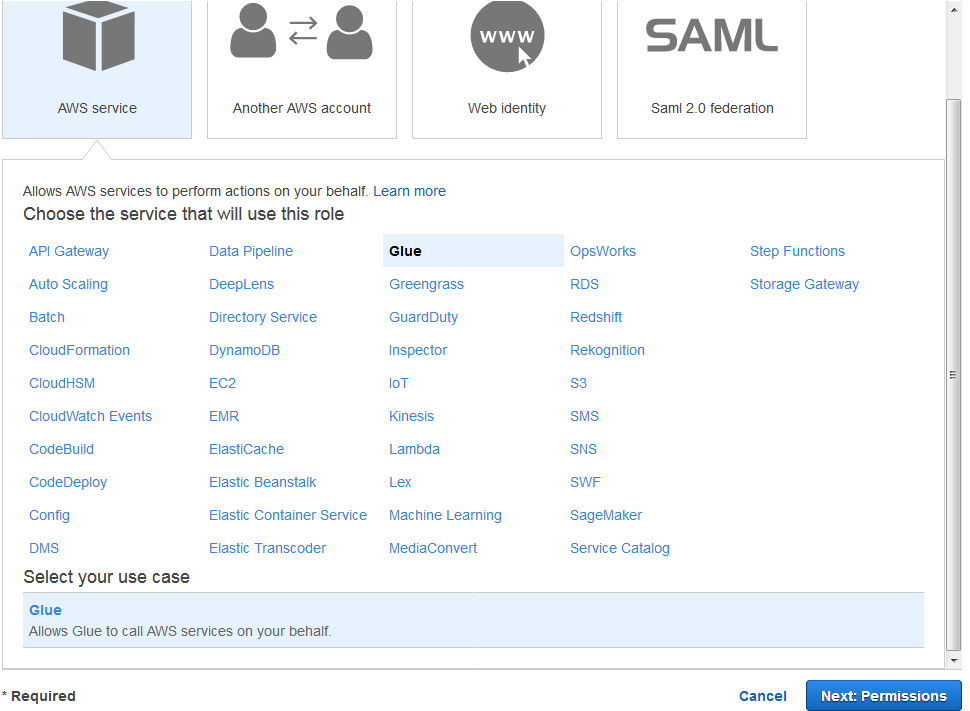


# Populate the AWS Glue Data Catalog

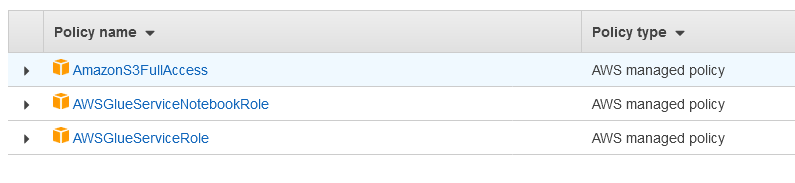
1. The AWS Glue Data Catalog is an index to the location, schema, and runtime metrics of your data. It contains references to data that is used as sources and targets of your extract, transform, and load (ETL) jobs in AWS Glue. The Data Catalog is a drop-in replacement for the Apache Hive Meta-store and provides a uniform repository where disparate systems can store and find metadata to keep track of data, and use that metadata to query and transform the data. To populate the data catalog, we need to first create a role with proper permissions and then a crawler to take inventory of the data in our S3 bucket.
2. Please use Chrome or Firefox browser to ensure smooth lab experience.
3. Sign into the AWS Management Console <https://console.aws.amazon.com/>.
4. In the upper-right corner of the AWS Management Console, confirm you are in the desired AWS region (e.g., N. Virginia).
5. Click on **IAM** from the list of all services. This will bring you to the IAM dashboard page.
6. Click **Roles** on the left hand panel and then click **Create role**



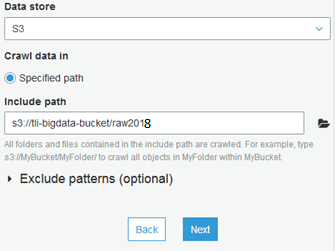
1. Select **Glue**



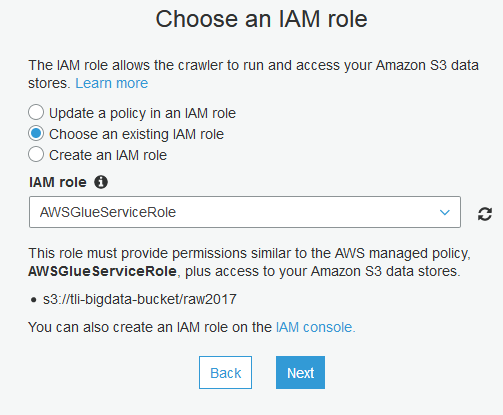
1. Click **Next: Permissions**
2. In attach permission policies page, add **AWSGlueServiceRole**, **AWSGlueSErviceNotebookRole**, and **AmazonS3FullAccess**.



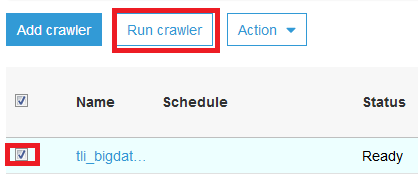
1. Click **Next:Review**
2. Name it **YourInitialsGlueServiceRole** and Click **Create Role**
3. Go back to AWS Management Console <https://console.aws.amazon.com/>.
4. In the upper-right corner of the AWS Management Console, confirm you are in the desired AWS region (e.g., N. Virginia).
5. Click on **Glue** from the list of all services. This will bring you to the AWS Glue dashboard page.
6. Click on **Crawlers** on the left panel and then click **Add crawler**
7. For Crawler name, enter **YourInitials\_bigdata\_raw\_stream**
8. Click **Next**
9. For Data store, ensure **S3** is selected. Choose **Specified path in my account** and for Include path, enter **s3://YourInitials-bigdata-bucket/raw2018**



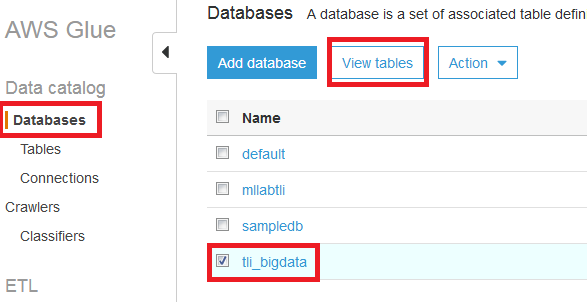
1. Click **Next**
2. Choose **No** to Add another data store
3. Click **Next**
4. **Choose an existing IAM role** and select **YourInitialsGlueServiceRole** in the drop down box



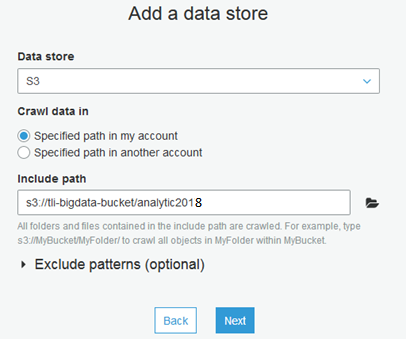
1. Click Next
2. For Frequency, choose **Run on demand** and click **Next**
3. For Database, click **Add database**, name it **YouInitials\_bigdata**, and click **Create**.
4. Click **Next**
5. Review the configuration and click **Finish**.
6. On the Crawlers page, tick the checkbox of the crawler just created and click **Run crawler**



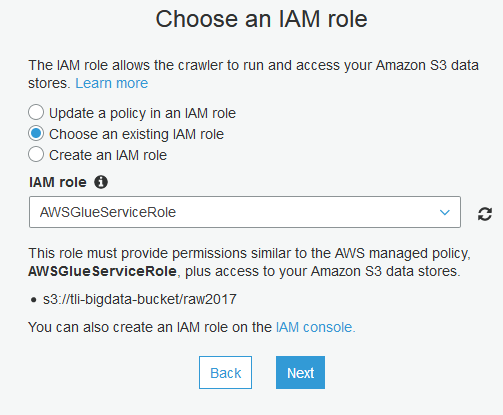
1. Wait for the crawler to finish.
2. Click Databases on the left panel and tick the checkbox next to **YourInitials\_bigdata** database, then click **View tables**.

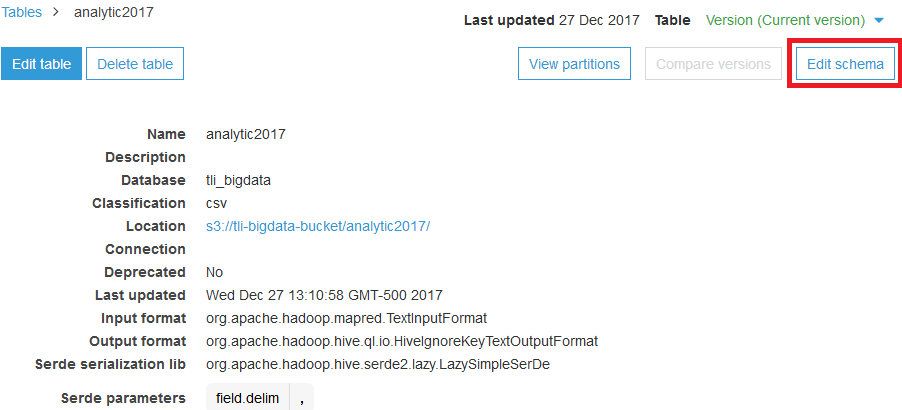


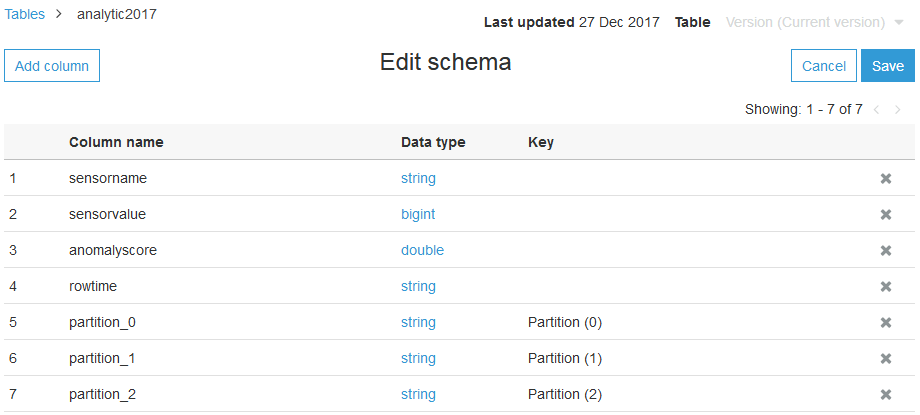
1. Click on the name of the table and review the configuration and schema created by the crawler. Notice the crawler was able to recognize the data format was json and it automatically partitioned the table based on the folder structure. A correct partition strategy can boost performance and save costs during querying as it helps reduce the amount of data to scan. Next we will add another crawler to index the data processed by Kinesis Analytics.
2. Click on **Crawlers** on the left panel and then click **Add crawler**
3. For Crawler name, enter **YourInitials\_bigdata\_analytic\_stream**
4. Click **Next**
5. For Data store, ensure **S3** is selected. Choose **Specified path in my account** and for Include path, enter **s3://YourInitials-bigdata-bucket/analytic2018**



1. Click **Next**
2. Choose **No** to Add another data store
3. Click **Next**
4. **Choose an existing IAM role** and select **YouInitialsGlueServiceRole** in the drop down box



1. Click Next
2. For Frequency, choose **Run on demand** and click **Next**
3. For Database, select the database with **YouInitials\_bigdata**.
4. Click **Next**
5. Review the configuration and click **Finish**.
6. On the Crawlers page, tick the checkbox for **YourInitials\_bigdata\_analytic\_stream** and click **Run crawler**
7. Wait for the crawler to finish.
8. Click Databases on the left panel and tick the checkbox next to **YourInitials\_bigdata** database, then click **View tables**.
9. Click on **analytic2018** table and notice the column names are not auto populated. The reason is that unlike JSON, we lost the column headers when we exported it in CSV with Kinesis Firehose. Let’s edit the schema ourselves.
10. Click on **Edit schema **
11. Enter the following
    * Col0: **sensorname**
    * Col1: **sensorvalue**
    * Col2: **anomalyscore**
    * Col3: **rowtime**

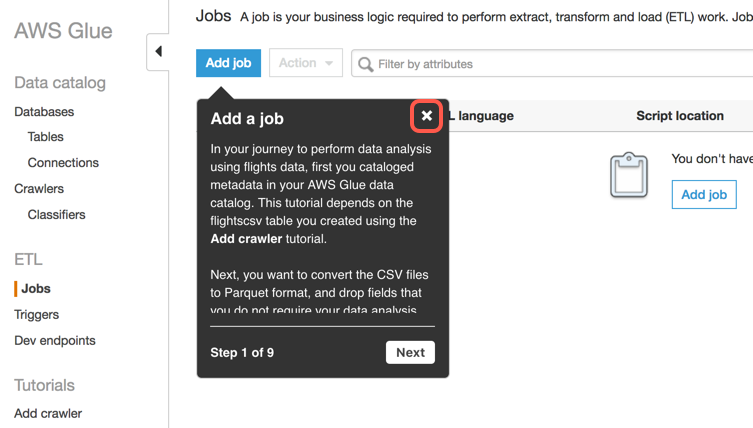


1. Click **Save**

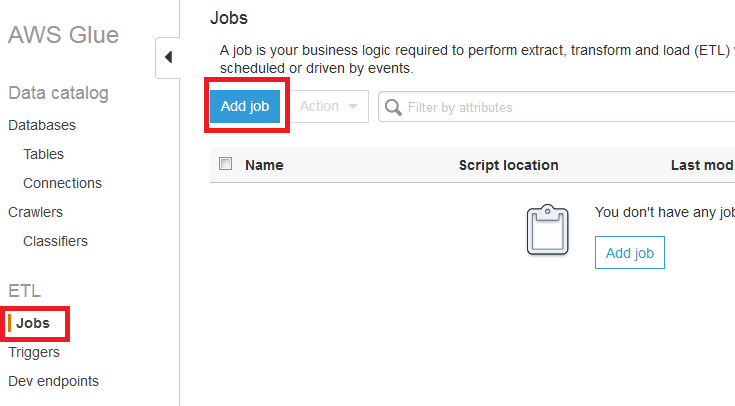
# 

# Transform Data with AWS Glue

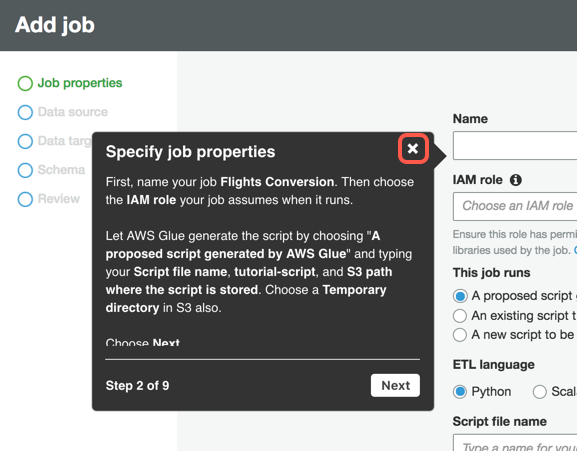
1. Recall in the first lab, data processed by Amazon Kinesis Analytics and then stored on S3 were CSV format. While we selected GZIP compression, CSV is not always the best format to store data in a big data environment, especially if each row has many columns. As part of best practice, we should optimize the data in columnar data stores such as Apache Parquet and Apache ORC. The columnar store formats can store data efficiently by employing column-wise compression, different encoding, compression based on data type, and predicate pushdown. They are also splittable. Generally, better compression ratios or skipping blocks of data means reading fewer bytes from S3 and leads to better query performance. In this section, we will leverage AWS Glue transforms the CSV data into Parquet.
2. Sign into the AWS Management Console <https://console.aws.amazon.com/>.
3. In the upper-right corner of the AWS Management Console, confirm you are in the desired AWS region (e.g., N. Virginia).
4. Click on **Glue** from the list of all services. This will bring you to the AWS Glue dashboard page.
5. Click on **Jobs** on the left panel.
6. Close the **Add a job** dialog box with instructions by clicking the “X” button in the upper right of the dialog.



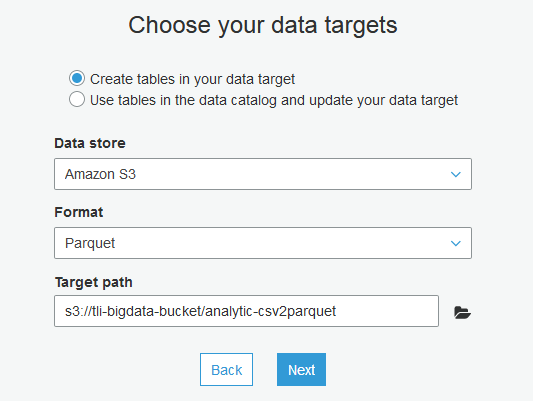
1. Click **Add job.**



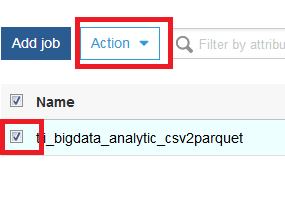
1. Close the **Specify job properties** dialog box with instructions by clicking the “X” button in the upper right of the dialog.



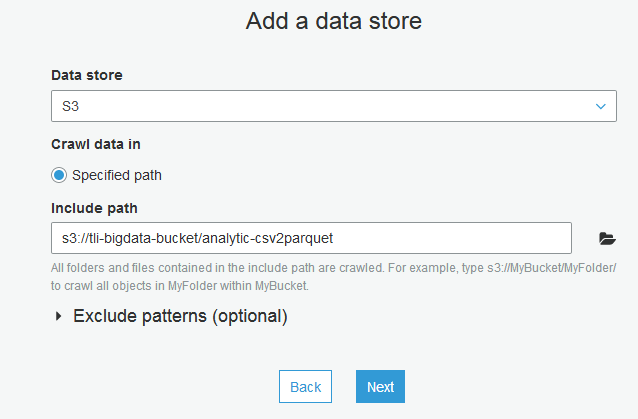
1. In Job properties page, enter the following
   * Name: **YourInitials\_bigdata\_analytic\_csv2parquet**
   * IAM role: **YouInitialsGlueServiceRole**
   * The job runs: **A proposed script generated by AWS Glue**
   * ETL Language: **Python**
   * Script file name: **YourInitials\_bigdata\_analytic\_csv2parquet**
   * S3 path where the script is stored:
   * Temporary directory: Copy and paste the S3 bucket path from S3 path above and append **/temp** behind the path. For example, s3://aws-glue-scripts-1234567890-us-east-1/**YourInitials(or root)/**temp
   * Expand Script libraries and job parameters section and change Concurrent DPUs per job run from **10** to **100**. This will help speed up the transformation process.
   * Leave everything else default
2. Click **Next**
3. Select **analytic2018** table and click **Next**
4. In Data target page, choose **Create tables in your data target**. Select **Amazon S3** as the Data store and **Parquet** as the Format. For Target path, use the same big data S3 bucket for the lab and append **analytic-csv2parquet**. For example, s3://**YourInitials**-bigdata-bucket/analytic-csv2parquet



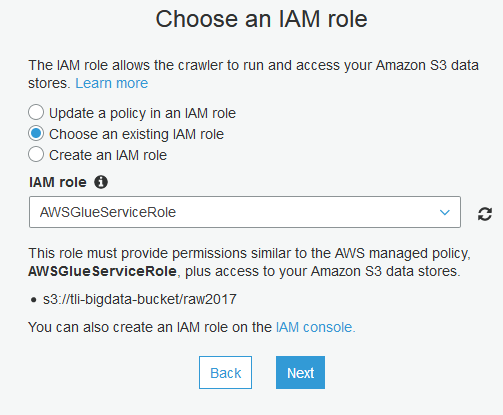
1. Click **Next**
2. In field mapping page, leave everthing default and click **Next**
3. Review the configuration and click **Finish**
4. In the Glue ETL editor page, review the code auto-genatered and click **Save**. Click on the **X** button on the top right hand corner to exit the code editor page.
5. Back in Jobs page, tick the job checkbox and click Action then select Run Job.



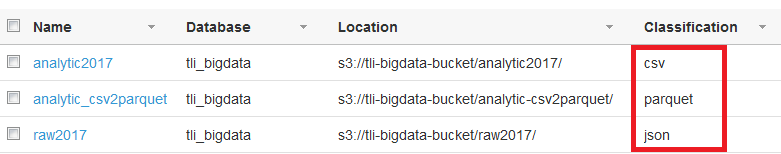
1. Take a break, this could take over 10 minutes to complete. Once complete, we will create one last crawler to index the new Parquet files.
2. Click on **Crawlers** on the left panel and then click **Add crawler**
3. For Crawler name, enter **YourInitials\_bigdata\_analytic\_parquet**
4. Click **Next**
5. For Data store, ensure **S3** is selected. Choose **Specified path in my account** and for Include path, enter **s3://YourInitials-bigdata-bucket/analytic-csv2parquet**



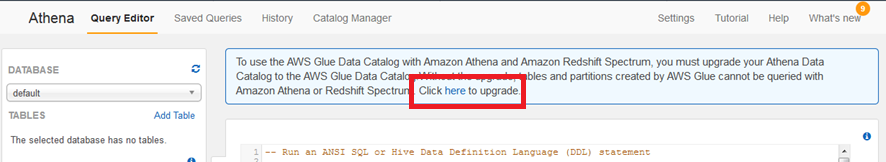
1. Click **Next**
2. Choose **No** to Add another data store
3. Click **Next**
4. **Choose an exiting IAM role** and select **YouInitialsGlueServiceRole** in the drop down box



1. Click Next
2. For Frequency, choose **Run on demand** and click **Next**
3. For Database, select the database with **YouInitials\_bigdata**.
4. Click **Next**
5. Review the configuration and click **Finish**.
6. On the Crawlers page, tick the checkbox for **YourInitials\_bigdata\_analytic\_parquet** and click **Run crawler**
7. Wait for the crawler to finish.
8. Click Databases on the left panel and tick the checkbox next to **YourInitials\_bigdata** database, then click View tables. Note AWS Glue Data Catalog is a very flexible and robust tool categorize big data in different format. With the tables schema defined we can now query the data with big data tools such as Athena, EMR and even Redshift Spectrum.



# Query Data with Amazon Athena and AWS Glue Data Catalog

1. Sign into the AWS Management Console <https://console.aws.amazon.com/>.
2. In the upper-right corner of the AWS Management Console, confirm you are in the desired AWS region (e.g., N. Virginia).
3. Click on **Athena** from the list of all services. This will bring you to the Amazon Athena dashboard page.
4. In the Database drop down, select **YourInitials\_bigdata** database. **Important Note**: If you do not see the database, it means Athena is not integrated with Glue Data Catalog and you will need to upgrade Athena to use it. Click on the link in the message box in Athena shown in screenshot below to perform the upgrade then continue to next step.
5. Run the following query in the query editor, it will return all sensors with anomaly score higher than 2

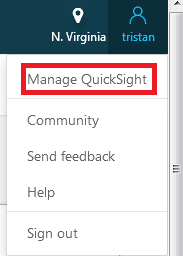
select sensorname, sensorvalue, anomalyscore from analytic\_csv2parquet where anomalyscore > 2 limit 10

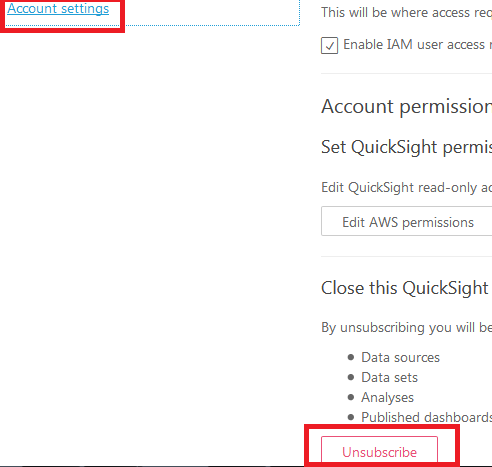
1. Feel free to run queries against other tables available to explore the data.

# Visualize Data with Amazon QuickSight and Amazon Athena

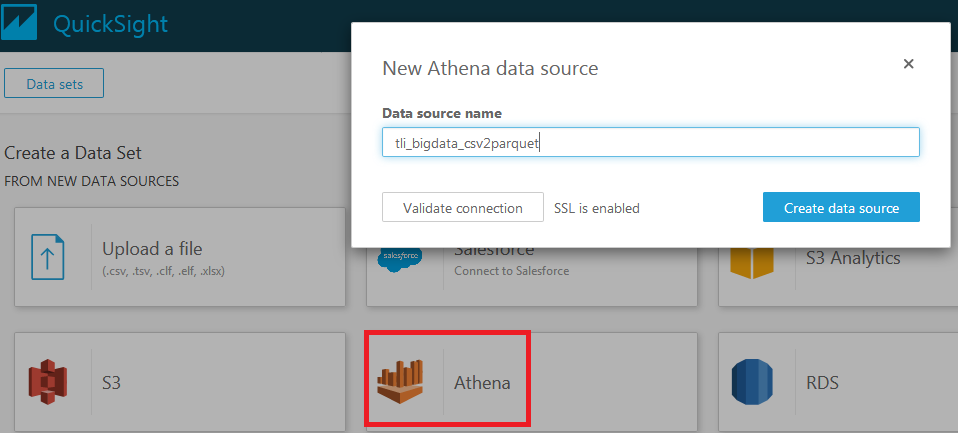
1. Sign into the AWS Management Console <https://console.aws.amazon.com/>.
2. In the upper-right corner of the AWS Management Console, confirm you are in the desired AWS region (e.g., N. Virginia).
3. Click on **QuickSight** from the list of all services. This will bring you to the Amazon QuickSight dashboard page.

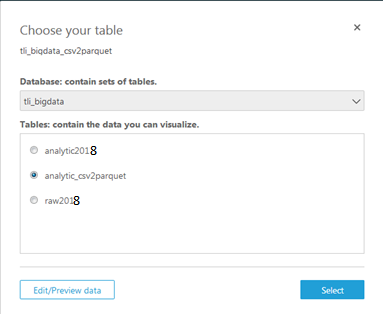
Note, if it’s your first time using QuickSight, go ahead and sign up with basic subscription. You can always unsubscribe in the account settings at the top right hand corner.



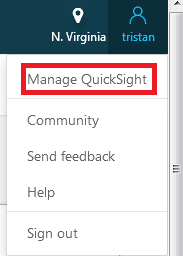


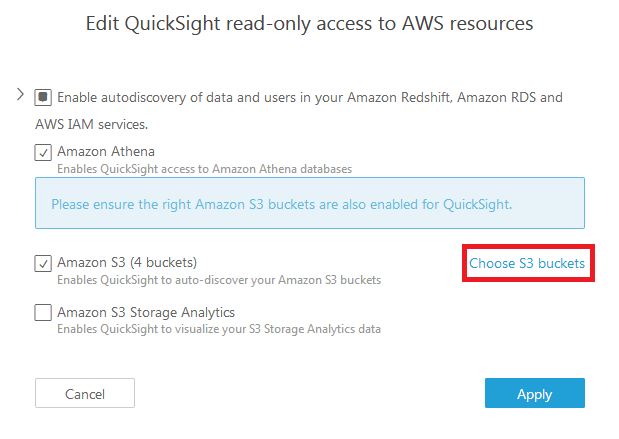
1. Click **Manage Data** in the top right hand corner area
2. Click **New data set**
3. In new data set page, click Athena and name it **YourInitials\_bigdata\_csv2parquet**



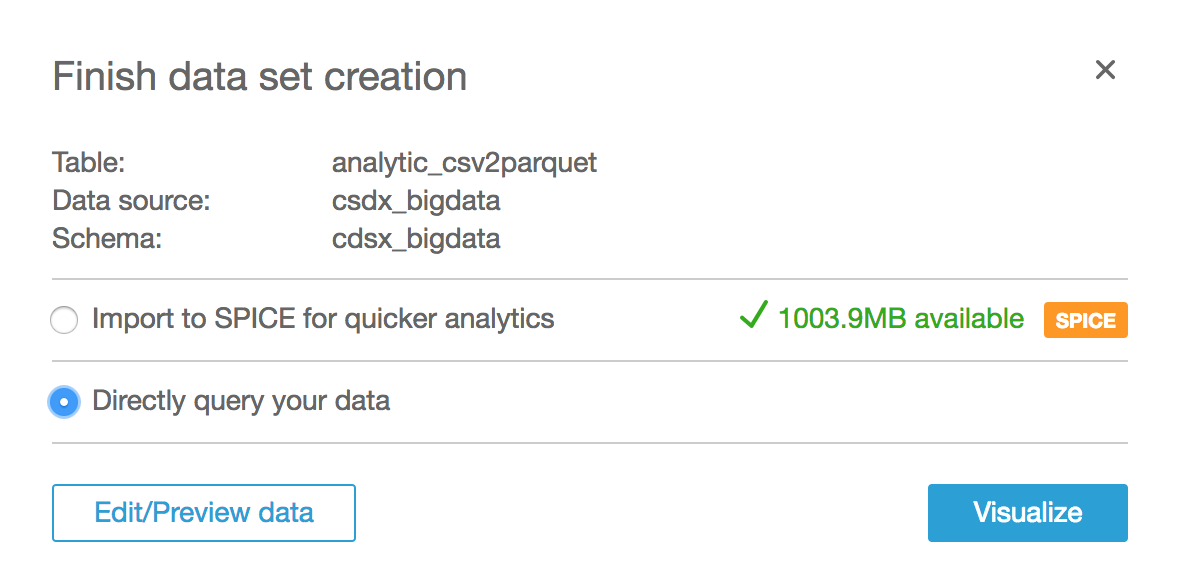
1. Click **Create data source**
2. Select **YourInitial\_bigdata** database and select **analytic\_csv2parquet** table 
3. Click **Select**.

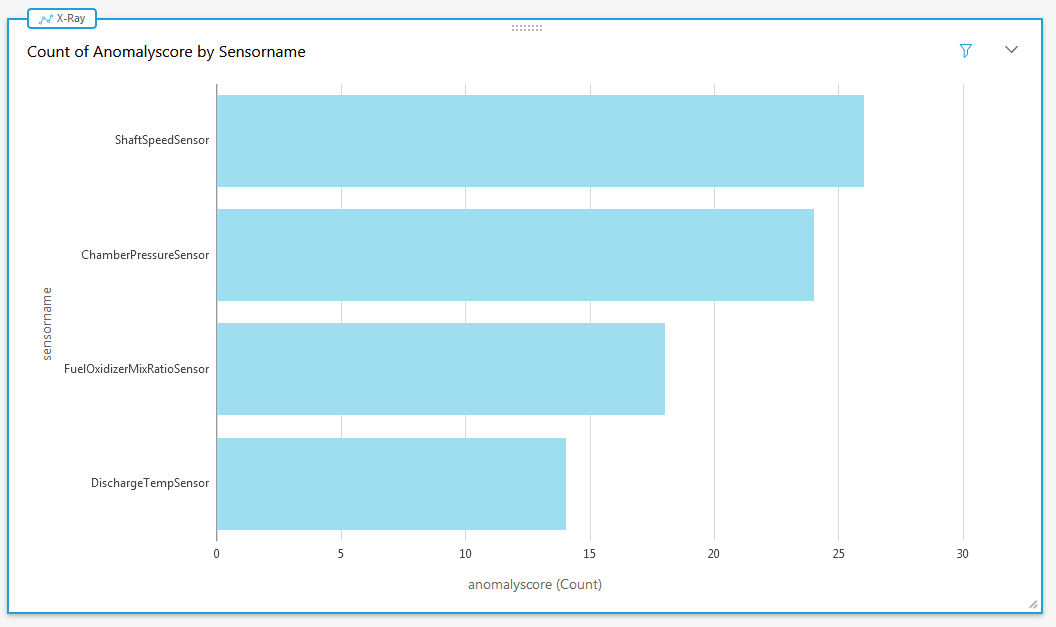
Note if you get a permission error, make sure to give QuickSight read-only permission in Account Settings to load data from the source S3 bucket.



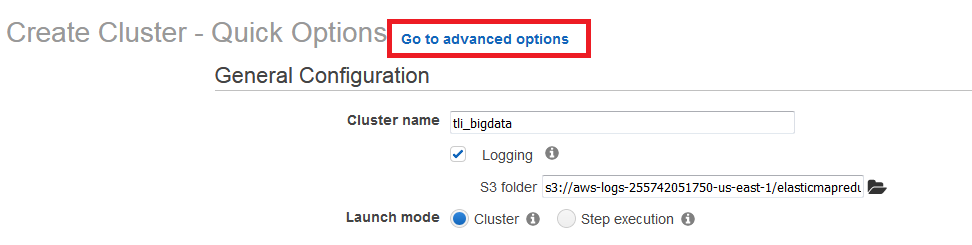
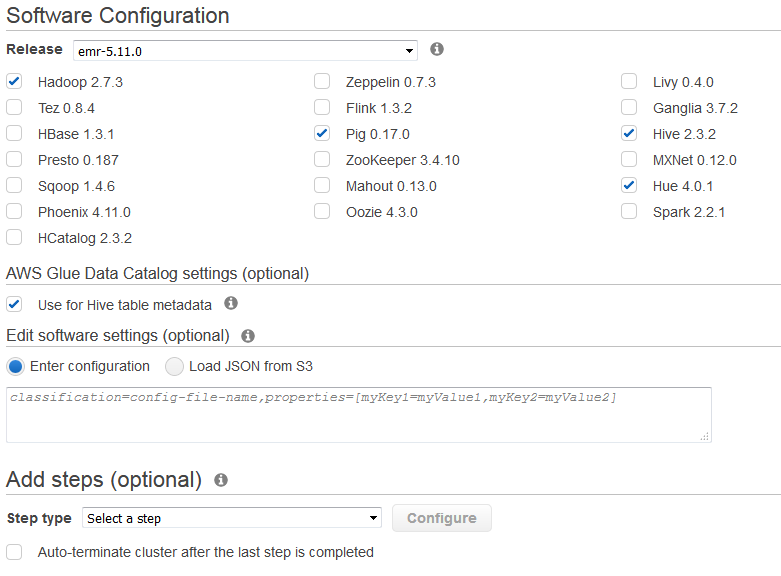


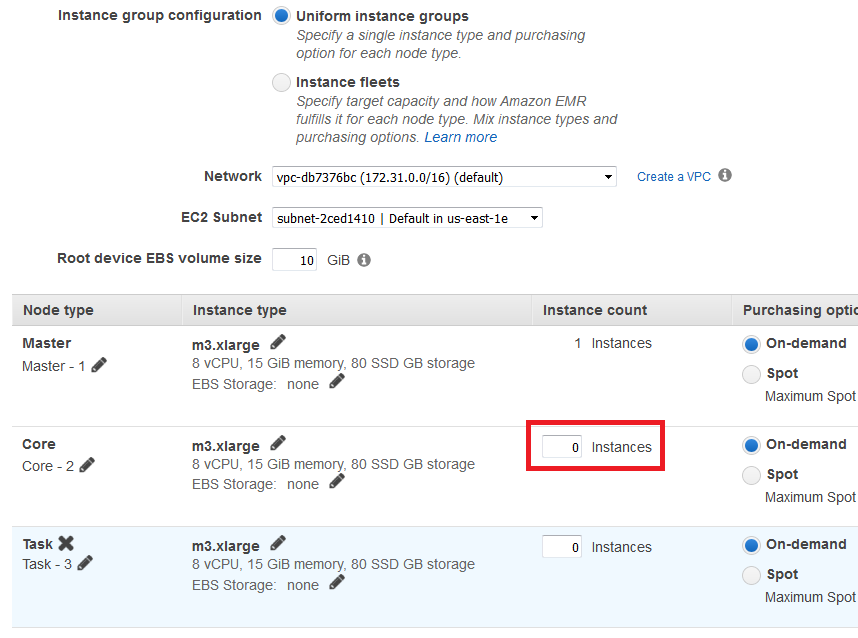
Note if you get a message asking to import into SPICE or query directly, choose to query directly, then click **Visualize**.

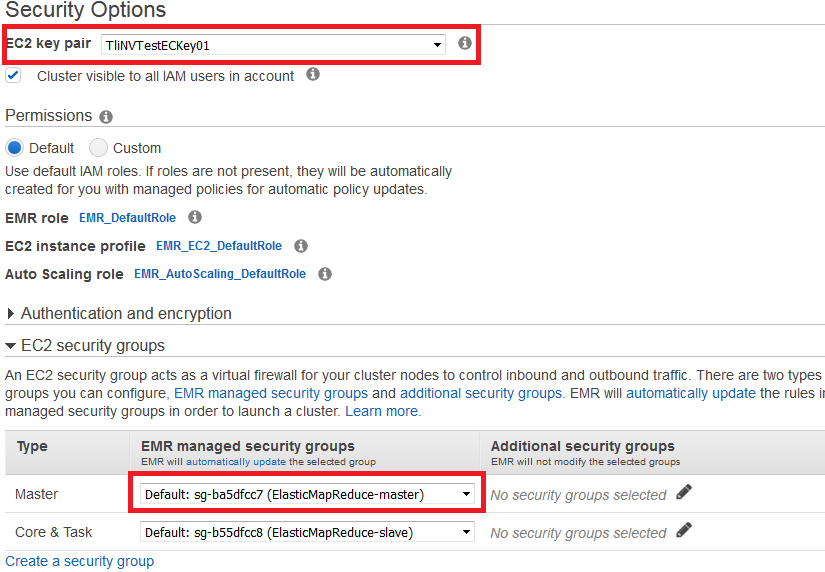


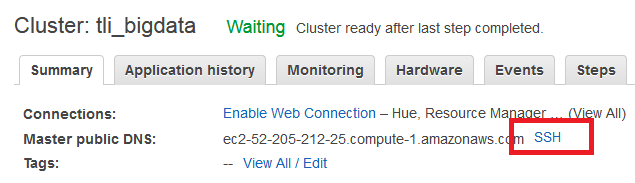
1. In the Finish data set creation page, leave everything default and click **Visualize**
2. In the graph designer page, feel free to drag and drop different metrics and graph types to build desired graphs. For example, the following graph shows the count of anomalies detected for each sensor. 

# Query Data with Amazon EMR and AWS Glue Data Catalog (Optional)

1. The AWS Glue Data Catalog is a drop-in replacement for the Apache Hive Meta-store, so it is also easy to run Amazon EMR (Elastic MapReduce) with Hive and query the data on S3 with Glue Data Catalog. In this section, we will create an EMR cluster to demonstrate how easy it really is.
2. Sign into the AWS Management Console <https://console.aws.amazon.com/>.
3. In the upper-right corner of the AWS Management Console, confirm you are in the desired AWS region (e.g., N. Virginia).
4. Click on **EMR** from the list of all services. This will bring you to the Amazon EMR dashboard page
5. Click **Create cluster**
6. Click **Go to advanced options** at the top of the page
7. For Release, select **emr-5.11.0**. Tick **Hadoop, Pig, and Hive and Hue** checkboxes. Tick **Use for Hive table metadata** checkbox, this is important because it enables us to access the Glue Data Catalog and query the data on S3 with Hive. 
8. Click **Next**
9. For this lab, we only need 1 node cluster, so go ahead and change Core instances from **2** to **0**. For Network and EC2 subnet, make sure to pick a VPC and subnet that is publicly accessible as we will SSH into the master node later. Note, if you run into capacity limit error to run m3.xlarge instance, change instance type to m4.xlarge by clicking on the pencil icon next to m3.xlarge in Master node type.



1. Click **Next**
2. For Cluster name, enter **YourInitials\_bigdata**. Leaving everything else default and click **Next**
3. Select an EC2 key pair that you have private PEM file and make sure Master security group allows inbound SSH port from public (anywhere).If you do not have a EC2 key pair, refer to Appendix below for instructions to create one.
4. Click **Create cluster**
5. Take a break and wait until the cluster status shows **Waiting**
6. Click on the EMR cluster with name **YourInitials\_bigdata**. In Summary page, click on SSH link.



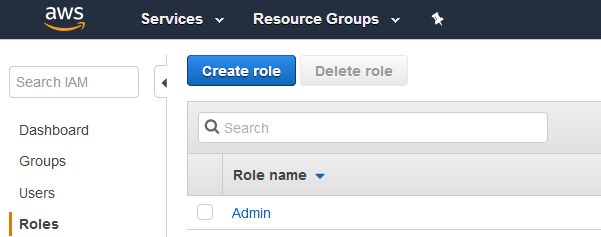
1. Follow the instruction and then SSH into the master node**.** Note, if you have issues connecting to the instance, make sure security group is opened to allow SSH inbound connection.
2. Enter **hive** in the terminal
3. Run the following HiveQL query, make sure to change database name with your initials

select sensorname, sensorvalue, anomalyscore from **YourInitial\_bigdata**.analytic\_csv2parquet where anomalyscore > 2 limit 10;

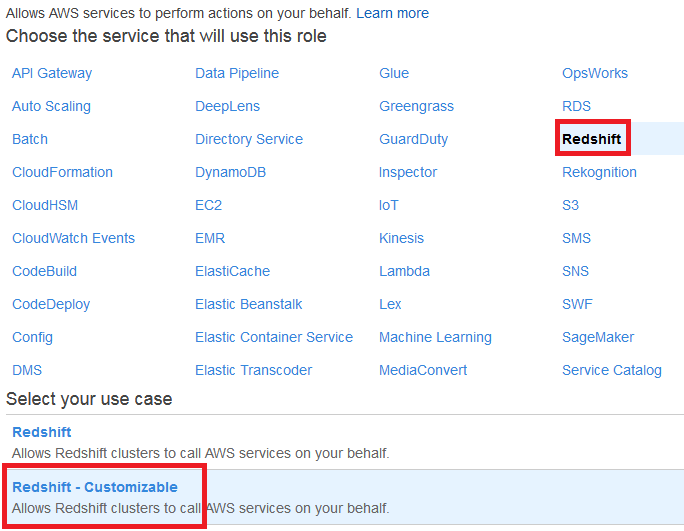
1. Feel free to run queries against other tables available to explore the data.

# Query Data with Amazon Redshift Spectrum and AWS Glue Data Catalog (Optional)

1. Note this section by no means a proper Redshift workshop, we are only going to focus on one of the Redshift features, Spectrum to learn how to query data in external datasource through AWS Glue Data Catalog.
2. **Prerequisites –** In order to connect and work with Amazon Redshift, you need a SQL query client such as SQL Workbench/J with JDBC or ODBC for RedShift installed. You can download and install Workbench/J from <http://www.sql-workbench.net>. Follow links provided for instructions on how to install and configure [JDBC](http://docs.aws.amazon.com/redshift/latest/mgmt/configure-jdbc-connection.html%22%20%5Co) and [ODBC](http://docs.aws.amazon.com/redshift/latest/mgmt/configure-odbc-connection.html%22%20%5Co%20%22Configure%20an%20ODBC%20Connection) for Redshift.
3. First we need create a role for Redshift Spectrum to have access to Glue Data Catalog and S3 resources.
4. Click on **IAM** from the list of all services. This will bring you to the IAM dashboard page.
5. Click **Roles** on the left hand panel and then click **Create role**



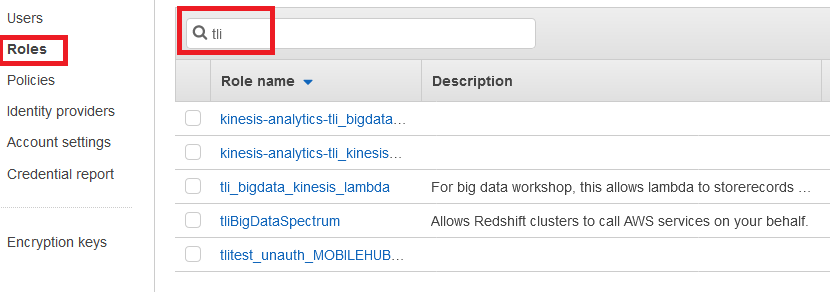
1. Select **Redshift** and select **Redshift - Customizable** for your use case



1. Click **Next: Permissions**
2. In attach permission policies page, add **AWSGlueServiceNotebookRole** and **AmazonS3FullAccess**.



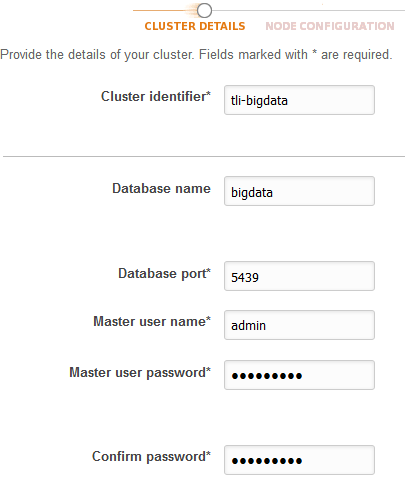
1. Click **Next: Review**
2. Name it **YourInitialsBigDataSpectrumRole** and Click **Create Role**
3. In the IAM roles, page search for the new role you created



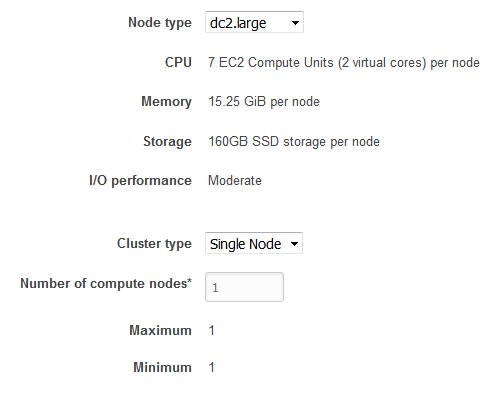
1. Click **YourInitialsBigDataSpectrumRole**



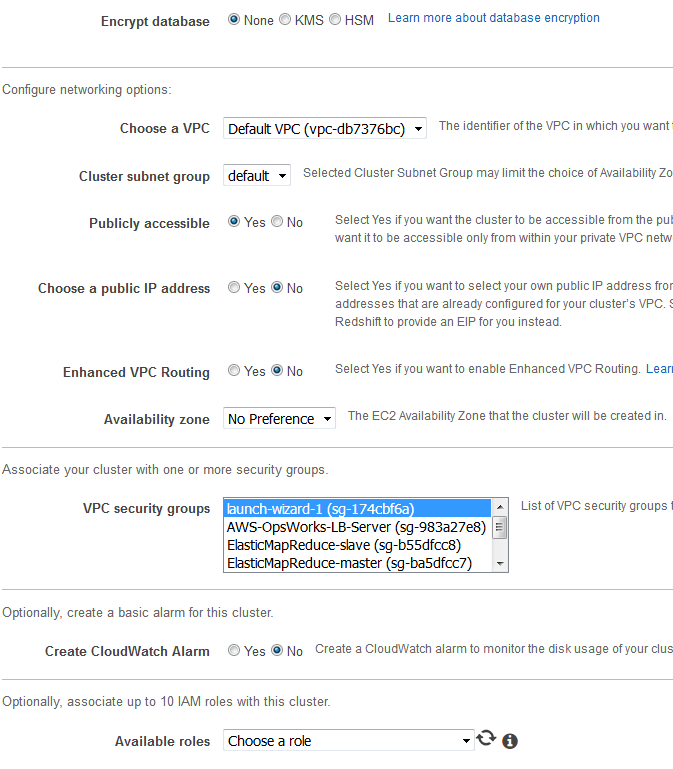
1. Copy the **Role ARN** for later use
2. Back to the AWS Management Console <https://console.aws.amazon.com/>.
3. In the upper-right corner of the AWS Management Console, confirm you are in the desired AWS region (e.g., N. Virginia).
4. Click on **Redshift** from the list of all services. This will bring you to the Amazon Redshift dashboard page
5. Click **Launch cluster**
6. In Cluster Details page, enter the following
   * Cluster identifier: **YourInitials-bigdata**
   * Database name: **bigdata**
   * Database port: **5439**
   * Master user name: **admin**
   * Master user password: **YourPassword**
   * Confirm password: **YourPassword**



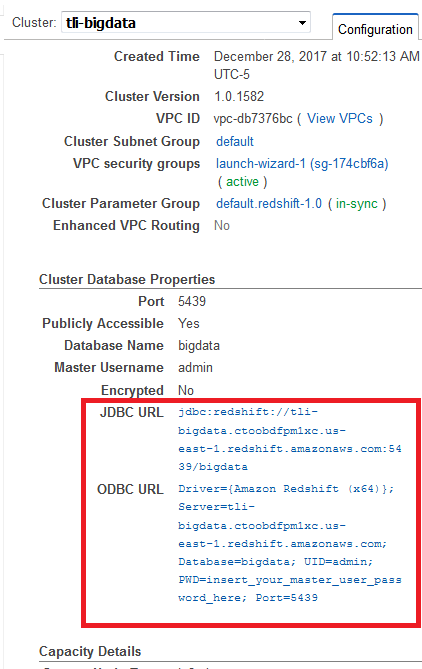
1. Click **Continue**
2. Leave everything default, single node dc2.large cluster is enough for this exercise.



1. Click **Continue**
2. In Additional Configuration page, enter the following
   * Cluster parameter group: **default**
   * Encrypt database: **None**
   * Choose a VPC: **Pick a VPC with Public Access**
   * Cluster subnet group: **default**
   * Publicly accessible: **Yes**
   * Choose a public IP address: **No**
   * Enhanced VPC Routing: **No**
   * Availablity zone: **No Preference**
   * VPC security groups: **Pick one,** make sure the selected security group allows inbound database port 5439 from public (anywhere).
   * Create CloudWatch Alarm: **No**
   * Available roles: **YourInitialsBigDataSpectrumRole**



1. Click **Continue**
2. Review configuration and click **Launch**
3. Take a break and wait until the cluster status shows **available**
4. Click YourInitals-bigdata cluster
5. In the detail page, copy the JDBC or ODBC URL and use it to connect to the Redshift cluster with your SQL query client such as Workbench/J.



1. Once connected with your SQL query client. Execute the following SQL, note to change region if you are not using N. Virginia region.

create external schema **YourInitial**\_bigdata from data catalog

database '**YourInitial\_bigdata**'

iam\_role '**Role ARN Copied in Step 12'**

region 'us-east-1';

1. You should receive a message indicating YourInitials\_bigdata schema created successfully. Execute the following SQL

select sensorname, sensorvalue, anomalyscore from **YourInitial\_bigdata**.analytic\_csv2parquet where anomalyscore > 2 limit 10;

# Conclusion

In this lab, you have learned how to leverage AWS Glue, Amazon Athena, Amazon EMR, Amazon Redshift, and Amazon QuickSight to continue to process, transform, query, analyze and visualize the data stored in your S3 data lake. In the next lab, you will learn how to use Amazon SageMaker to train the stored data, then host the model, and finally calling the SageMaker Endpoint for prediction/inference.

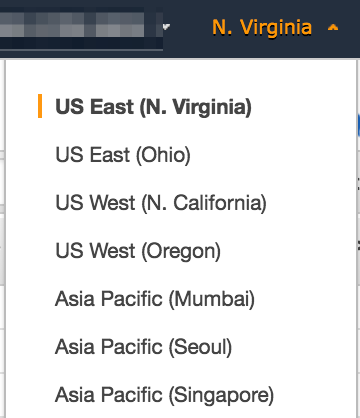
# Appendix

**Create New EC2 Key Pair**

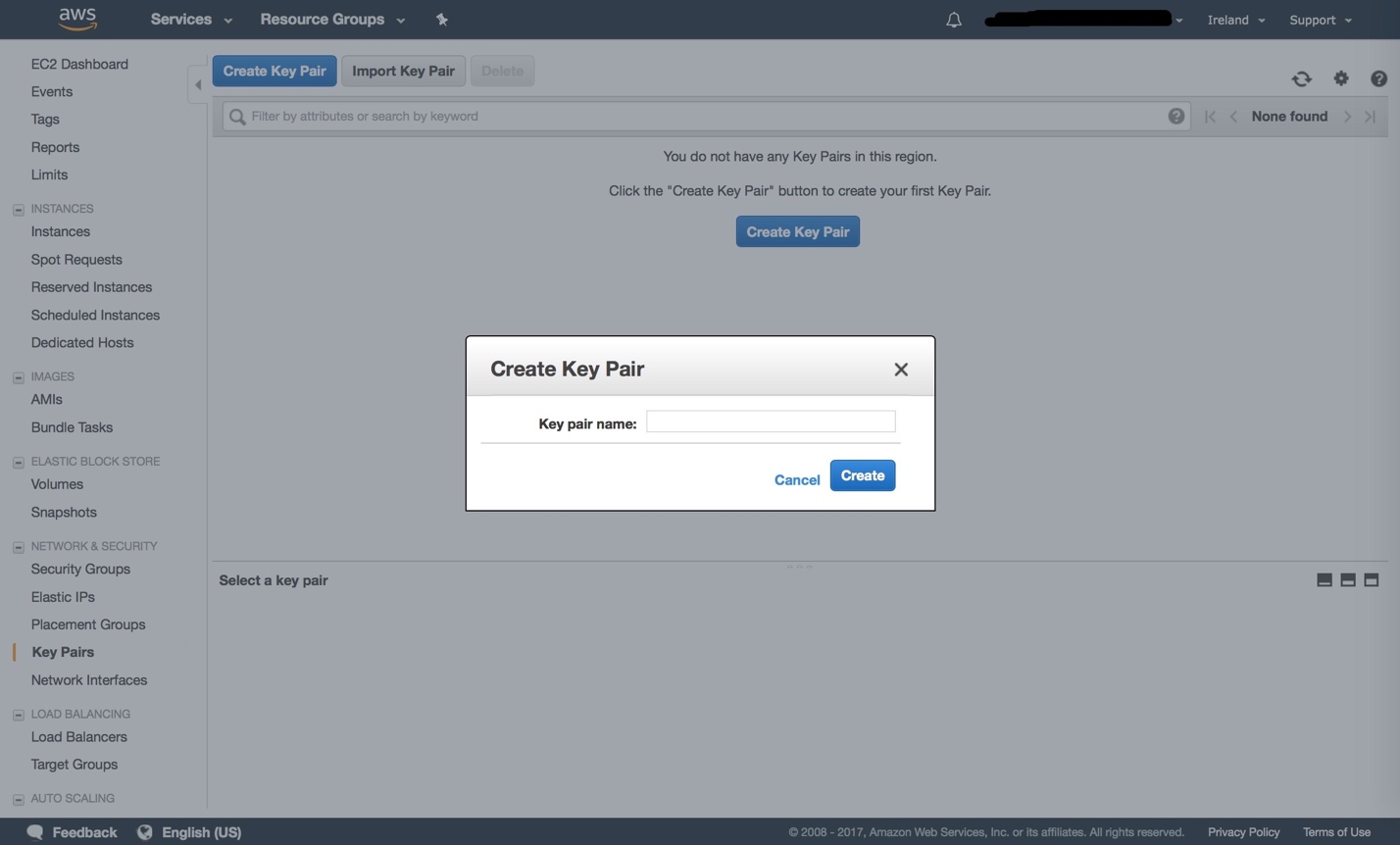
Before launching CloudFormation, you will need to create an EC2 Key Pair which will be used to allow administrative access into the EC2 Instances created by CloudFormation.

Sign in to the AWS management console. Open the EC2 page by clicking the searching for EC2 or going to <https://console.aws.amazon.com/ec2/home>.

Once on the EC2 page, look in the upper right corner for the AWS Region and change this to **US East (N. Virginia)**



* Under **Network & Security**, select **Key Pairs** on the left and select **Create Key Pair**. Recommended name “DMSlab”. Please append your initials if multiple people are running in the same account, such as “DMSlab\_ws”.



* + Mac users: You will receive a download of a pem.txt file. Rename the file and remove the .txt to make it a .pem file. For example: DMSlab\_ws.pem. Note where this file was downloaded.
  + Windows users: You will be prompted to download a .pem file. Save this, and note where this was downloaded.

You have completed creating a new EC2 Key Pair.