



Amazon Web Services Data Engineering Immersion Day

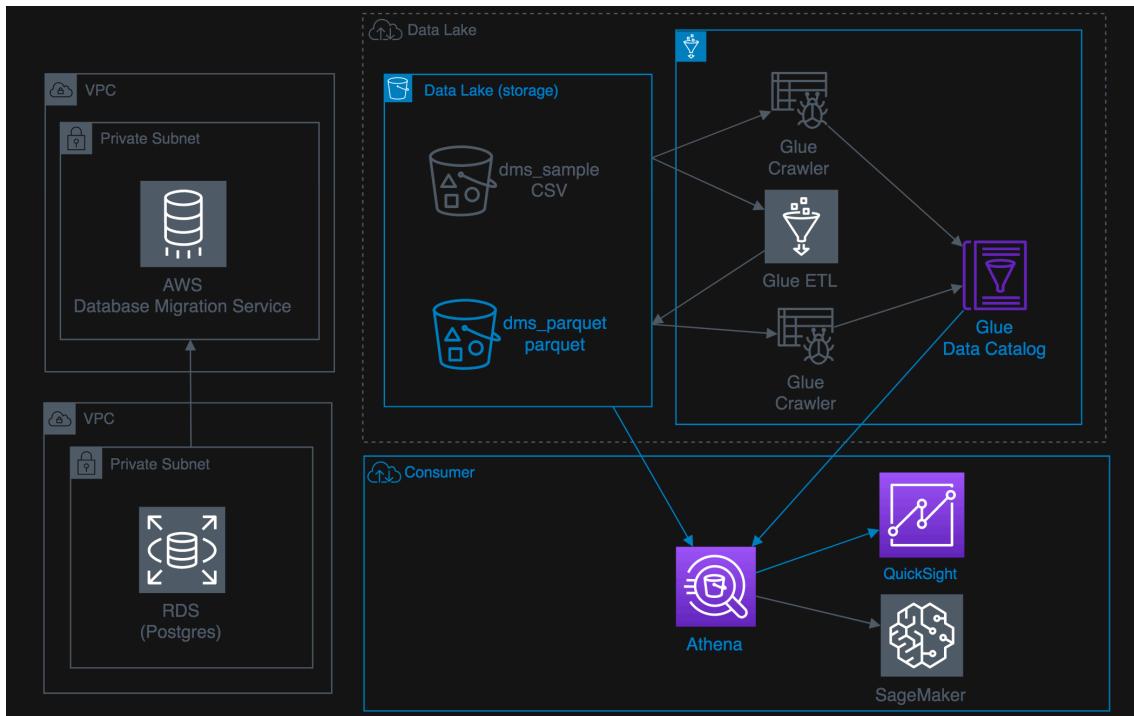
Exploring Data Lake with Amazon Athena and Amazon
QuickSight
March 2020

Table of Contents

<i>Introduction</i>	2
Prerequisites.....	2
Getting Started.....	2
<i>Query Data with Amazon Athena</i>	3
<i>Build an Amazon QuickSight Dashboard</i>	11
Set up QuickSight.....	11
Create QuickSight Charts.....	18
Create QuickSight Parameters.....	21
Create a QuickSight Filter.....	24
Add Calculated Fields.....	25
<i>Amazon QuickSight ML-Insights (Optional)</i>	28
<i>Athena Workgroups to Control Query Access and Costs (Optional)</i>	30
Workflow setup to separate workloads	30
Explore the features of workgroups	33
Managing Query Usage and Cost.....	39
Cost Allocation Tags.....	45

Introduction

This lab introduces you to AWS Glue, Amazon Athena, and Amazon QuickSight. AWS Glue is a fully managed data catalog and ETL service; Amazon Athena queries data; and Amazon QuickSight provides visualization of the data you import.



Prerequisites

The DMS Lab and Glue ETL lab is a prerequisite for this lab.

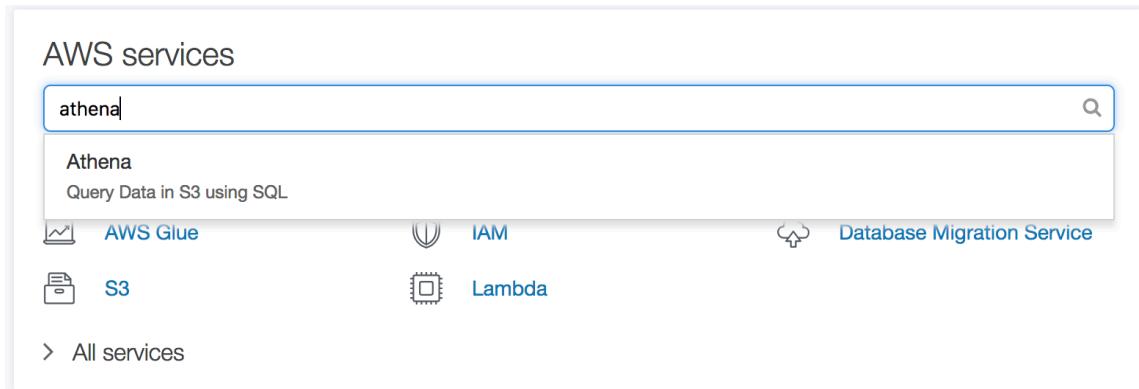
Getting Started

In this lab, you will complete the following tasks:

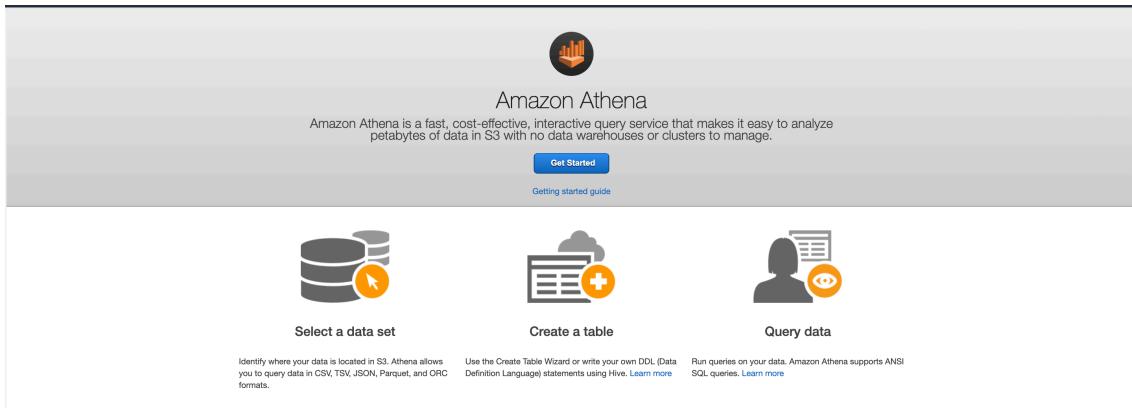
1. [Query data and create a view with Amazon Athena](#)
2. [Athena Workgroups to Control Query Access and Costs](#)
3. [Build a dashboard with Amazon QuickSight](#)

Query Data with Amazon Athena

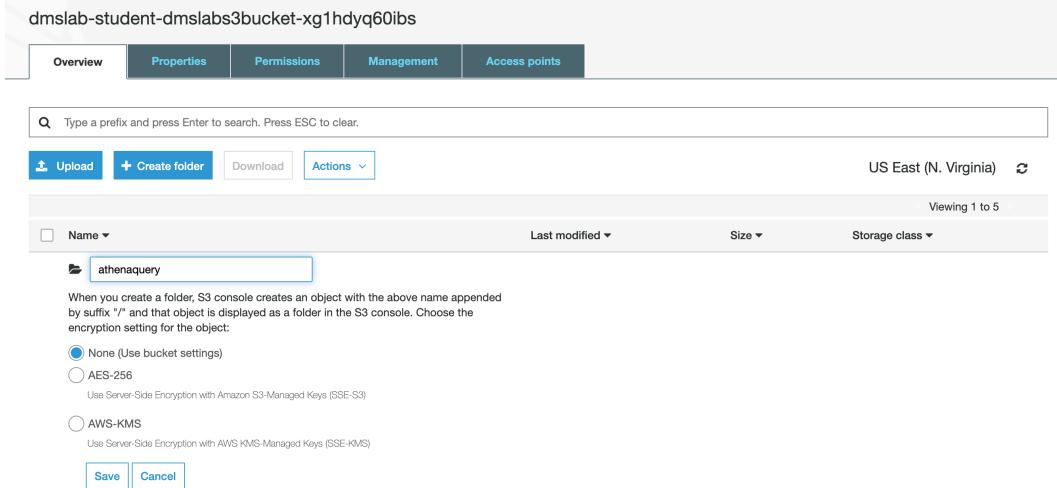
1. In the AWS services console, search for **Athena**.



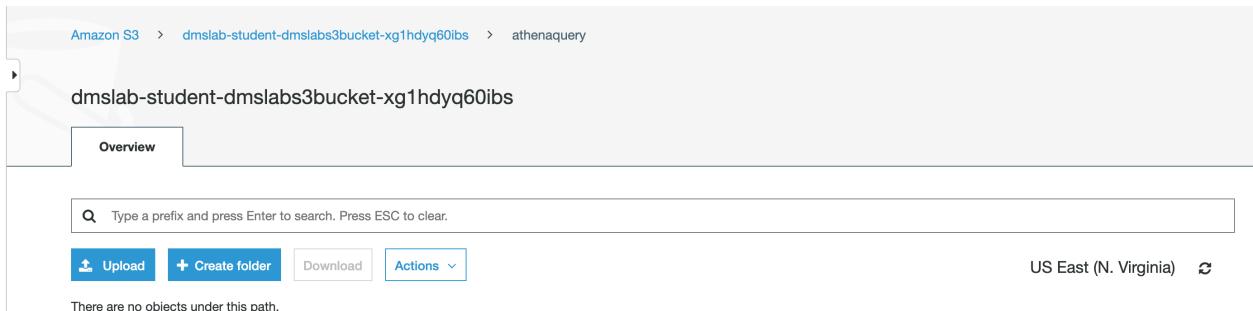
2. If you are using Athena first time, click on "Get Started" button in introduction screen.



3. Setup the S3 buckets to store the query results. For this, Navigate to S3 console, click on bucket created as part of student lab (e.g: <dmslab-student-dmslabs3bucket-xg1hdqyq60ibs>). Inside bucket, click on "create folder". Name folder as **athenaquery**. Click on **Save**.



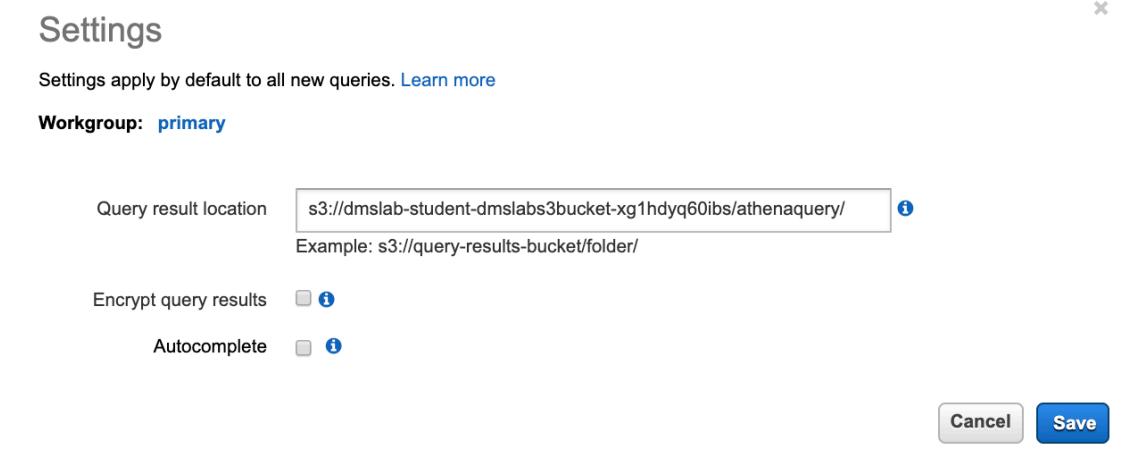
4. Note down the path of S3 folder created above and save it.



5. In Athena Console, click on **Settings**.

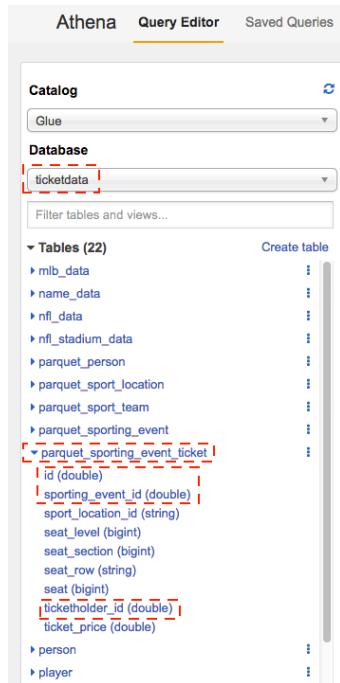


6. Copy paste the folder path as shown below:



7. In the Query Editor, select your newly created database e.g., "**ticketdata**".
8. Click the table named "**parquet_sporting_event_ticket**" to inspect the fields.

Note: The type for fields id, sporting_event_id and ticketholder_id should be (double).



Next, we will query across tables `parquet_sporting_event`, `parquet_sport_team`, and `parquet_sport location`.

9. Copy the following SQL syntax into the New Query 1 tab and click **Run Query**.

```
SELECT
    e.id AS event_id,
    e.sport_type_name AS sport,
    e.start_date_time AS event_date_time,
    h.name AS home_team,
    a.name AS away_team,
    l.name AS location,
    l.city
FROM parquet_sporting_event e,
     parquet_sport_team h,
     parquet_sport_team a,
     parquet_sport_location l
WHERE
    e.home_team_id = h.id
    AND e.away_team_id = a.id
    AND e.location_id = l.id;
```

The results appear beneath the query window.

```

SELECT
    e.event_id AS event_id,
    e.sport_type_name AS sport,
    h.name AS home_team,
    a.name AS away_team,
    l.name AS location,
    l.city
FROM parquet_sporting_event e,
     parquet_sport_team h,
     parquet_sport_team a,
     parquet_sport_location l
WHERE e.home_team_id = h.id
      AND e.away_team_id = a.id
      AND e.location_id = l.id

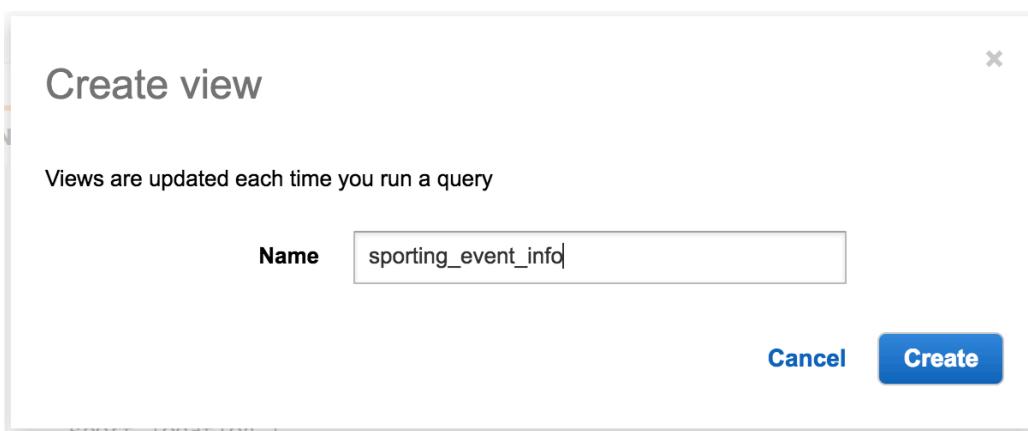
```

Run query Save as Create (Run time: 2.79 seconds, Data scanned: 12.67 KB)
 Use Ctrl + Enter to run query, Ctrl + Space to autocomplete
 Create table from query
 Create view from query
 Format query Clear

	event_id	sport	event_date_time	home_team	away_team	location	city
1	9101	football	2020-09-14 19:00:00.000	Kansas City Chiefs	San Diego Chargers	Arrowhead Stadium	Kansas City, Missouri
2	9111	football	2020-09-21 18:00:00.000	Kansas City Chiefs	Denver Broncos	Arrowhead Stadium	Kansas City, Missouri
3	9091	football	2020-09-19 13:00:00.000	Kansas City Chiefs	Oakland Raiders	Arrowhead Stadium	Kansas City, Missouri
4	8421	baseball	2020-04-10 00:00:00.000	Toronto Blue Jays	Detroit Tigers	Rogers Centre	Toronto Ontario
5	8431	baseball	2020-04-08 00:00:00.000	Toronto Blue Jays	Astoria Braves	Rogers Centre	Toronto Ontario
6	8451	baseball	2020-05-10 00:00:00.000	Toronto Blue Jays	Los Angeles Dodgers	Rogers Centre	Toronto Ontario
7	8461	baseball	2020-05-17 00:00:00.000	Toronto Blue Jays	Kansas City Royals	Rogers Centre	Toronto Ontario
8	8741	football	2020-09-14 19:00:00.000	Miami Dolphins	New York Jets	Hard Rock Stadium	Miami Gardens, Florida
9	8764	football	2020-09-14 19:00:00.000	Miami Dolphins	Biloxi Shuckers	Hard Rock Stadium	Miami Gardens, Florida

10. As shown above Click **Create** and then select **Create view from query**

11. Name the view "sporting_event_info" and click **Create**.



Your new view is created

Athena Query Editor Saved Queries History Data sources Workgroup : primary

Data source: awsdatacatalog Database: ticketdata

Tables (24)

Views (1)

sporting_event_info

event_id (bigint)
sport (string)
event_date_time (timestamp)
home_team (string)
away_team (string)
location (string)
city (string)

Connect data source

CREATE OR REPLACE VIEW "sporting_event_info" AS

```

SELECT
    e.event_id AS event_id,
    e.sport_type_name AS sport,
    h.name AS home_team,
    a.name AS away_team,
    l.name AS location,
    l.city
FROM parquet_sporting_event e,
     parquet_sport_team h,
     parquet_sport_team a,
     parquet_sport_location l
WHERE e.home_team_id = h.id
      AND e.away_team_id = a.id
      AND e.location_id = l.id

```

Run query Save as Create (Run time: 0.71 seconds, Data scanned: 0 KB)
 Use Ctrl + Enter to run query, Ctrl + Space to autocomplete

12. Copy the following SQL syntax into the New Query 2 tab and click on **Save Query**.

```

SELECT t.id AS ticket_id,
       e.event_id,
       e.sport,
       e.event_date_time,
       e.home_team,
       e.away_team,
       e.location,
       e.city,
       t.seat_level,
       t.seat_section,
       t.seat_row,
       t.seat,
       t.ticket_price,
       p.full_name AS ticketholder
FROM sporting_event_info e,
     parquet_sporting_event_ticket t,
     parquet_person p
WHERE
    t.sporting_event_id = e.event_id
    AND t.ticketholder_id = p.id

```

The screenshot shows a database query editor interface. At the top, there are tabs for "sporting_event_i..." and "New query 2". A blue "+" button is also visible. The main area contains a numbered SQL query:

```

1  SELECT t.id AS ticket_id,
2      e.event_id,
3      e.sport,
4      e.event_date_time,
5      e.home_team,
6      e.away_team,
7      e.location,
8      e.city,
9      t.seat_level,
10     t.seat_section,
11     t.seat_row,
12     t.seat,
13     t.ticket_price,
14     p.full_name AS ticketholder
15 FROM sporting_event_info e,
16      parquet_sporting_event_ticket t,
17      parquet_person p
18 WHERE
19     t.sporting_event_id = e.event_id
20     AND t.ticketholder_id = p.id

```

Below the query are three buttons: "Run query" (blue), "Save as" (grey), and "Create" (grey).

Click on **Save as** button Give this query a name:
create_view_sporting_event_ticket_info and some description and then, click on **Save**.

Choose a name

Name Use 1 - 128 characters

Description Use upto 1024 characters

Back to the query editor, you will see the query name changed. Now, click on Run Query.

sporting_event_i... **create_view_spor...**

```
1 SELECT t.id AS ticket_id,
2     e.event_id,
3     e.sport,
4     e.event_date_time,
5     e.home_team,
6     e.away_team,
7     e.location,
8     e.city,
9     t.seat_level,
10    t.seat_section,
11    t.seat_row,
12    t.seat,
13    t.ticket_price,
14    p.full_name AS ticketholder
15 FROM sporting_event_info e,
16     parquet_sporting_event_ticket t,
17     parquet_person p
18 WHERE
19     t.sporting_event_id = e.event_id
20     AND t.ticketholder_id = p.id
21
```

Run query **Save as** **Create ▾**

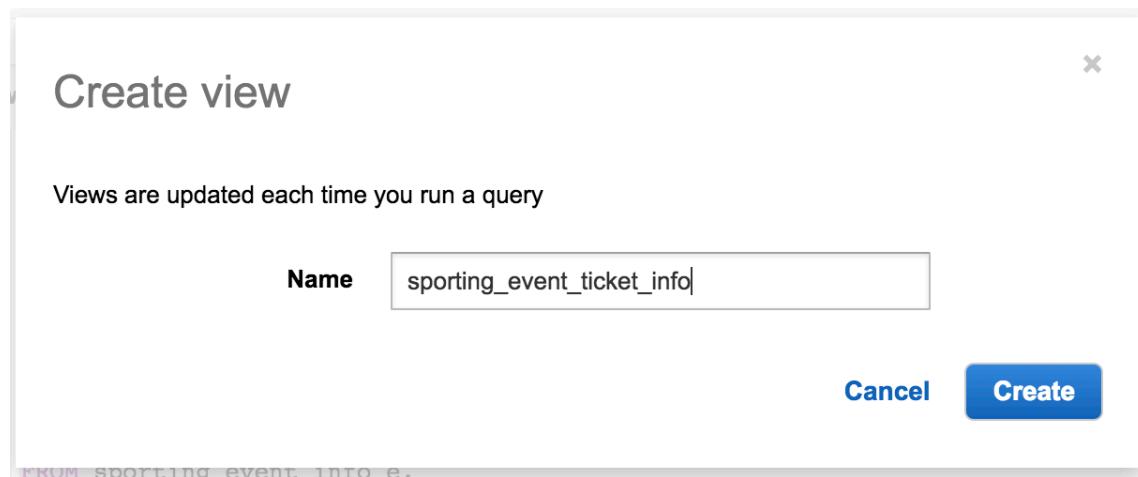
The results appear beneath the query window.

The screenshot shows the AWS Athena Query Editor interface. A query has been run, and the results are displayed in a table titled 'Results'. The table contains 10 rows of data, each representing a ticket record. The columns are: ticket_id, event_id, sport, event_date_time, home_team, away_team, location, city, seat_level, seat_section, seat_row, seat, ticket_price, and ticketholder. The data shows various football events at different locations like University of Phoenix Stadium and Sun Devil Stadium, with ticket prices ranging from \$10.00 to \$45.00.

	ticket_id	event_id	sport	event_date_time	home_team	away_team	location	city	seat_level	seat_section	seat_row	seat	ticket_price	ticketholder
1	7039811.0	59821	football	2020-11-18 18:00:00.000	Arizona Cardinals	Kansas City Chiefs	University of Phoenix Stadium	Gilbert, Arizona	3	17	B	2	45.00	Hossa Sauerz
2	7039811.0	59821	football	2020-11-18 18:00:00.000	Arizona Cardinals	Kansas City Chiefs	University of Phoenix Stadium	Gilbert, Arizona	3	17	B	3	45.00	Hossa Sauerz
3	7039811.0	59821	football	2020-11-18 18:00:00.000	Arizona Cardinals	Kansas City Chiefs	University of Phoenix Stadium	Gilbert, Arizona	3	17	A	3	45.00	Hossa Sauerz
4	7039811.0	59821	football	2020-11-18 18:00:00.000	Arizona Cardinals	Kansas City Chiefs	University of Phoenix Stadium	Gilbert, Arizona	3	17	A	4	45.00	Hossa Sauerz
5	7039811.0	59821	football	2020-11-18 18:00:00.000	Arizona Cardinals	Kansas City Chiefs	University of Phoenix Stadium	Gilbert, Arizona	3	17	A	1	45.00	Hossa Sauerz
6	7039811.0	59821	football	2020-11-18 18:00:00.000	Arizona Cardinals	Kansas City Chiefs	University of Phoenix Stadium	Gilbert, Arizona	3	17	A	2	45.00	Hossa Sauerz
7	7039811.0	59821	football	2020-11-18 18:00:00.000	Arizona Cardinals	Kansas City Chiefs	University of Phoenix Stadium	Gilbert, Arizona	3	16	B	2	45.00	Hossa Sauerz
8	7039811.0	59821	football	2020-11-18 18:00:00.000	Arizona Cardinals	Kansas City Chiefs	University of Phoenix Stadium	Gilbert, Arizona	3	16	B	1	45.00	Hossa Sauerz
9	7039811.0	59821	football	2020-11-18 18:00:00.000	Arizona Cardinals	Kansas City Chiefs	University of Phoenix Stadium	Gilbert, Arizona	3	16	B	3	45.00	Hossa Sauerz
10	7039811.0	59821	football	2020-11-18 18:00:00.000	Arizona Cardinals	Kansas City Chiefs	University of Phoenix Stadium	Gilbert, Arizona	3	16	B	4	45.00	Hossa Sauerz

13. As shown above Click **Create view from query**.

14. Name the view "**sporting_event_ticket_info**" and click **Create**.



15. Copy the following SQL syntax into the New Query 3 tab.

```
SELECT
    sport,
    count(distinct location) as locations,
    count(distinct event_id) as events,
    count(*) as tickets,
    avg(ticket_price) as avg_ticket_price
FROM sporting_event_ticket_info
GROUP BY 1
ORDER BY 1;
```

Click on **Save Query** and give this query name: **analytics_sporting_event_ticket_info** and some description and then, click on **Save**.

Choose a name

Name analytics_sporting_event_ticket_info
Use 1 - 128 characters

Description to analyze the view: sporting_event_ticket_info
Use upto 1024 characters

Cancel **Save**

The name of the New Query 3 will be changed to one assigned in previous step. Click on **Run Query**.

```

sporting_event_i... ✓ create_view_spor... ✎ + analytics_sportin... ✎
1 SELECT
2   sport,
3   count(distinct location) as locations,
4   count(distinct event_id) as events,
5   count(*) as tickets,
6   avg(ticket_price) as avg_ticket_price
7 FROM sporting_event_ticket_info
8 GROUP BY 1
9 ORDER BY 1;
10

```

Run query **Save as** **Create**

Your query returns two results in approximately five seconds. The query scans 25 MB of data, which prior to converting to parquet, would have been 1.59GB of CSV files.

The screenshot shows the AWS Athena Query Editor interface. At the top, there are tabs for 'Athena', 'Query Editor' (which is selected), 'Saved Queries', 'History', and 'AWS Glue Data Catalog'. Below the tabs, a list of saved queries is shown: 'New query 1', 'New query 2', 'New query 3', and a '+' button. The 'New query 3' entry is highlighted with an orange border. The main area contains the SQL query code:

```

1 SELECT
2   sport,
3   count(distinct location) as locations,
4   count(distinct event_id) as events,
5   count(*) as tickets,
6   avg(ticket_price) as avg_ticket_price
7 FROM sporting_event_ticket_info
8 GROUP BY 1
9 ORDER BY 1;
10

```

Below the code, there are three buttons: 'Run query', 'Save as', and 'Create'. A status message indicates '(Run time: 6.21 seconds, Data scanned: 25.97 MB)'. A note at the bottom says 'Use Ctrl + Enter to run query, Ctrl + Space to autocomplete'.

Results

	sport	locations	events	tickets	avg_ticket_price
1	baseball	30	294	958680	53.89345581425812
2	football	25	113	810304	57.40977502271104

The purpose of saving the queries is to have clear distinction between the results of the queries running on one view. Otherwise, your query results will be saved under “Unsaved” folder within the S3 bucket location provided to Athena to store query results. Please navigate to S3 bucket to observe these changes, as shown below:

The screenshot shows the AWS S3 console. The path is 'Amazon S3 > dmslab-student-dmslabs3bucket-xg1hdyq60ibs > athenaquery'. The 'Overview' tab is selected. A search bar at the top right contains the placeholder 'Type a prefix and press Enter to search. Press ESC to clear.' Below the search bar are buttons for 'Upload', '+ Create folder', 'Download', and 'Actions'. To the right, it shows the region 'US East (N. Virginia)' with a refresh icon. A table lists the objects in the folder:

Name	Last modified	Size	Storage class
analytics_sporting_event_ticket_info	--	--	--
create_view_sporting_event_ticket_info	--	--	--

At the bottom, a message 'Viewing 1 to 2' is displayed.

Build an Amazon QuickSight Dashboard

Set up QuickSight

1. In the AWS services console, search for **QuickSight**.

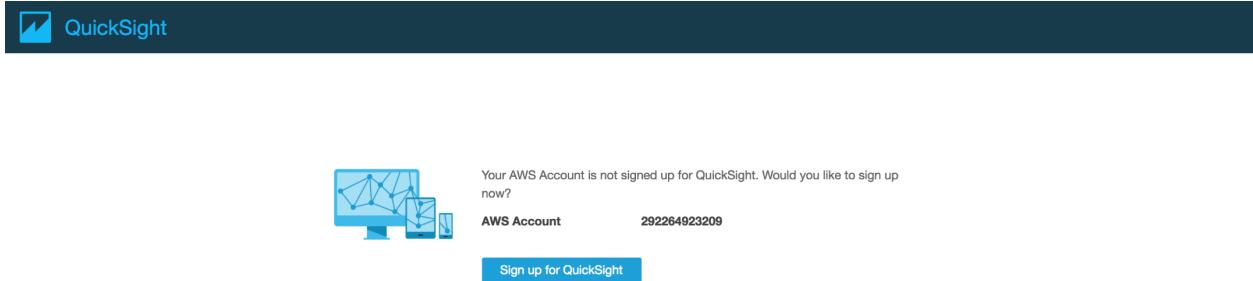
The screenshot shows the AWS services console. A search bar at the top contains the text 'QuickSight'. Below the search bar, the results are displayed:

- QuickSight**: Fast, easy to use business analytics
- Athena**
- IAM**
- S3**
- CloudWatch**
- AWS Glue**

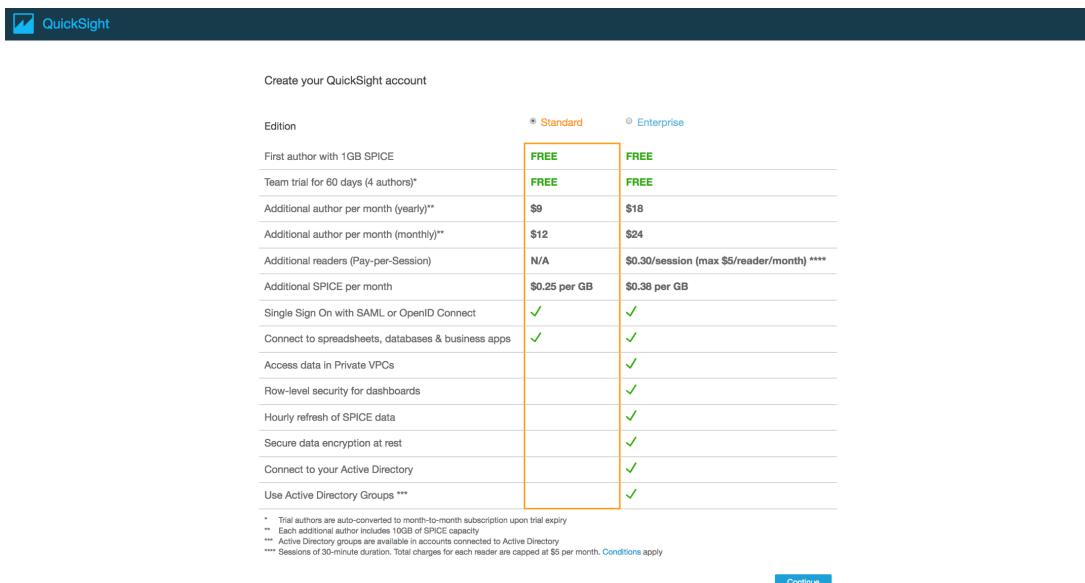
At the bottom left, there is a link 'All services'.

If this is the first time you have used QuickSight, you are prompted to create an account.

2. Click **Sign up for QuickSight**.



3. For account type, choose **Standard**. If you plan to complete the Bonus Exercise, please choose **Enterprise** Version
4. Click **Continue**.



5. On the Create your QuickSight account page, fill out your name and email address.
6. Select region and the check boxes to enable auto discovery, Amazon Athena, and Amazon S3.
7. Click **Choose S3 buckets** and select your DMS bucket (e.g., "dmslab-student-").
8. Click **Finish**.

Create your QuickSight account

Edition Standard

QuickSight account name Glue-Lab-George

You will need this for you and others to sign in.

Notification email address julbrigh+dataenglab@amazon.com

For QuickSight to send important notifications.

QuickSight capacity region US East (N. Virginia)

Select a region.

> Enable autodiscovery of data and users in your Amazon Redshift, Amazon RDS and AWS IAM services.

Amazon Athena
Enables QuickSight access to Amazon Athena databases

Please ensure the right Amazon S3 buckets are also enabled for QuickSight.

Amazon S3 (1 bucket)
Enables QuickSight to auto-discover your Amazon S3 buckets

Amazon S3 Storage Analytics
Enables QuickSight to visualize your S3 Storage Analytics data

Amazon IoT Analytics
Enable QuickSight to visualize your IoT Analytics data

Choose S3 buckets

Finish

9. On the **QuickSight landing page**, on the top right corner, click on “**Manage QuickSight**”.

Search for analyses, data sets, and dashboards

New analysis

All analyses All dashboards Favorites Tutorial videos

All analyses Last updated

Manage QuickSight

- Community
- Send feedback
- What's new

10. Choose “**Security and Permissions**” and under “**QuickSight access to AWS Services**, click on “**Add or Remove**” button.

Manage users

Your subscriptions

SPICE capacity

Account settings

Security & permissions

QuickSight can control access to AWS resources for the entire account in addition to individual users and groups

QuickSight access to AWS services

Amazon Redshift Amazon RDS IAM Amazon S3 Amazon Athena Amazon S3 Storage Analytics

AWS IoT Analytics

By configuring access to AWS services, QuickSight can access the data in those services. Access by users and groups can be controlled through the options below.

Add or remove

11. If you will observe there is an unchecked box against S3 buckets for "dmslab-student-", please check the box.



QuickSight access to AWS services

QuickSight can connect to the selected AWS products & services below for all users & groups:

	Amazon Redshift	<input checked="" type="checkbox"/>
	Amazon RDS	<input checked="" type="checkbox"/>

	IAM	<input checked="" type="checkbox"/>
	Amazon S3	<input type="checkbox"/>

Enables QuickSight to auto-discover your Amazon S3 buckets [Details](#)

	Amazon Athena	<input checked="" type="checkbox"/>
	Amazon S3 Storage Analytics	<input checked="" type="checkbox"/>

	AWS IoT Analytics	<input checked="" type="checkbox"/>
	Amazon SageMaker	<input type="checkbox"/>

12. Select Amazon S3. Select the S3 bucket created as part of student lab (for e.g: dmslab-student-dmslabs3bucket-xg1hdyq60ibs) which will have all the folders for your source data.

<input checked="" type="checkbox"/>	dmslab-student-dmslabs3bucket-xg1hdyq60ibs	<input type="checkbox"/>
-------------------------------------	--	--------------------------

13. Then, click on Finish.

Finish

14. You will observe that now there is a check mark in the checkbox for Amazon S3. This confirms that QuickSight has required permissions. Then, click on "Update".

QuickSight access to AWS services

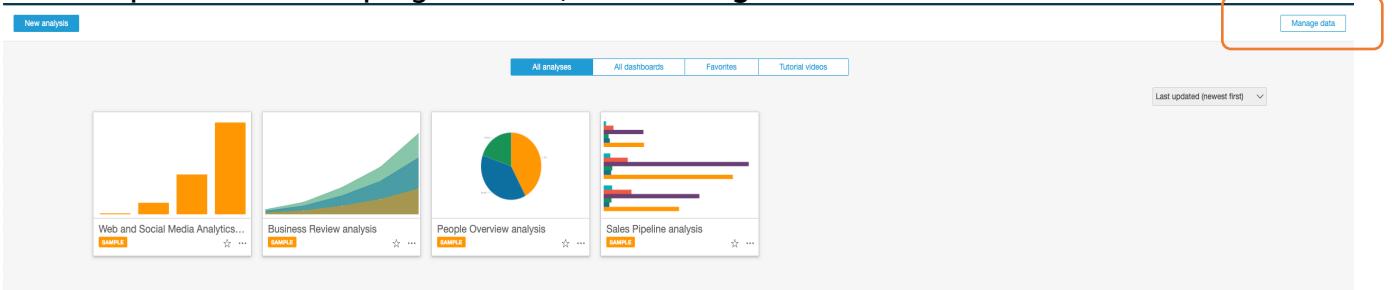
QuickSight can connect to the selected AWS products & services below for all users & groups:

	Amazon Redshift Enables QuickSight to auto-discover clusters	<input checked="" type="checkbox"/>
	Amazon RDS Enables QuickSight to auto-discover instances	<input checked="" type="checkbox"/>
	IAM Enables you to invite IAM users from this AWS Account to access QuickSight	<input checked="" type="checkbox"/>
	Amazon S3 Enables QuickSight to auto-discover your Amazon S3 buckets Details	<input checked="" type="checkbox"/>
	Amazon Athena Enables QuickSight access to Amazon Athena databases	<input checked="" type="checkbox"/>
	Amazon S3 Storage Analytics Enables QuickSight to visualize your S3 Storage Analytics data	<input checked="" type="checkbox"/>
	AWS IoT Analytics Enables QuickSight to visualize your IoT Analytics data	<input checked="" type="checkbox"/>
	Amazon SageMaker Enables QuickSight to infer fields from custom ML models	<input type="checkbox"/>

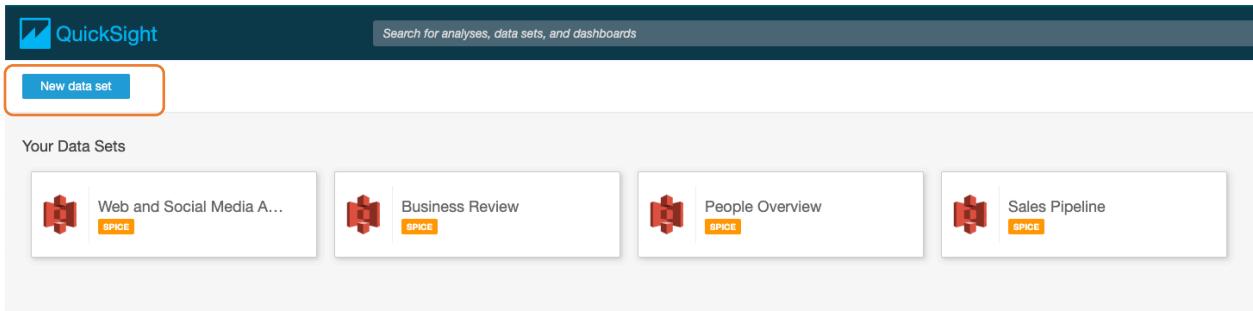
[Cancel](#)

[Update](#)

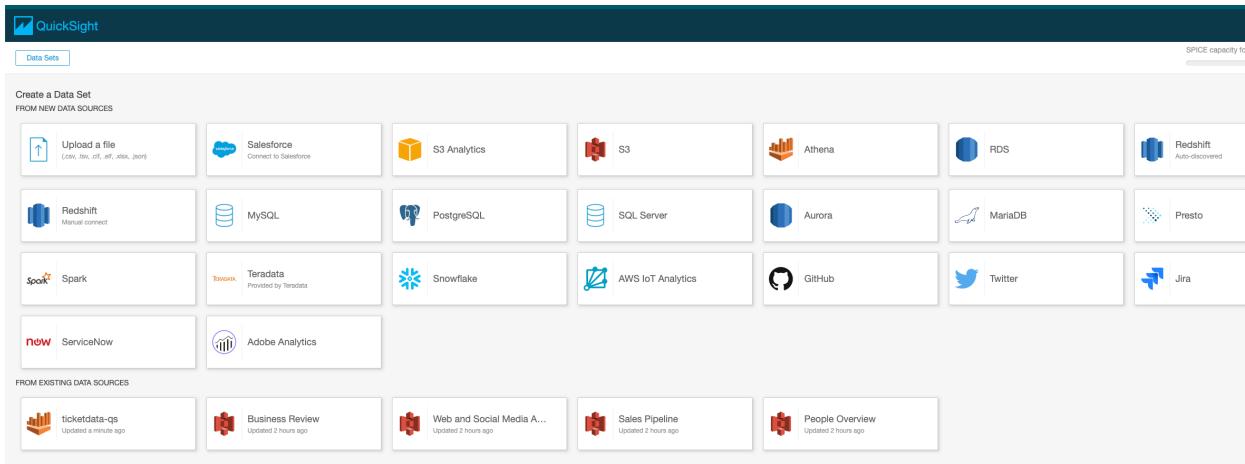
15. Navigate to QuickSight landing page by clicking on the QuickSight logo on the top left. On the top right corner, click **Manage Data.**



16. Click **New Data Set.**



17. On the Create a Data Set page, select **Athena** as the data source.



18. For Data source name, type "ticketdata-qs" and click **Validate connection**.

19. Click **Create data source**.

New Athena data source

Data source name
ticketdata-qs

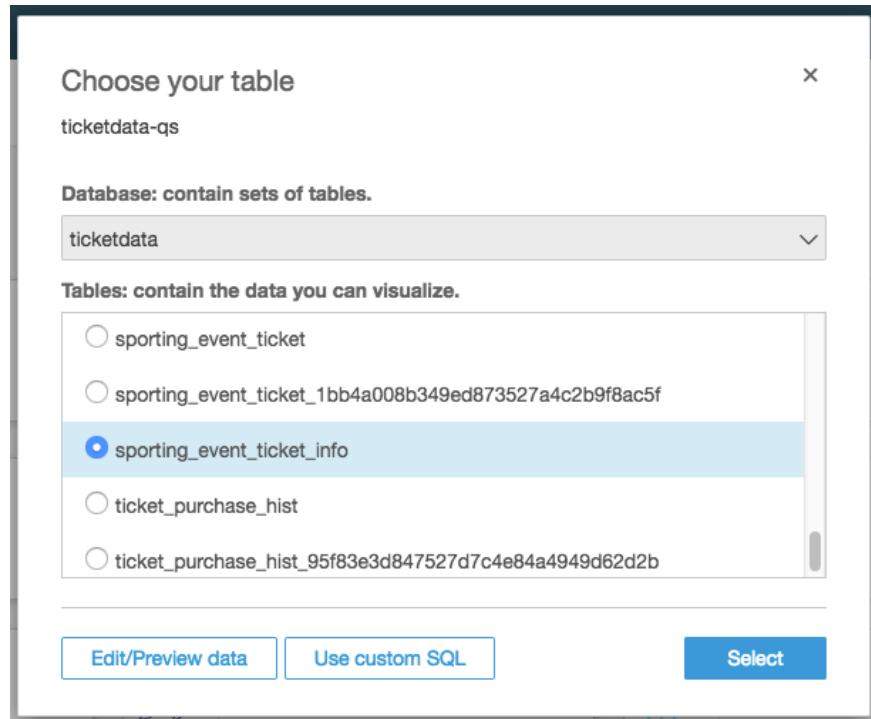
Athena workgroup
[primary]

Athena workgroup selection is now available for all Athena data sources. [Learn more](#)

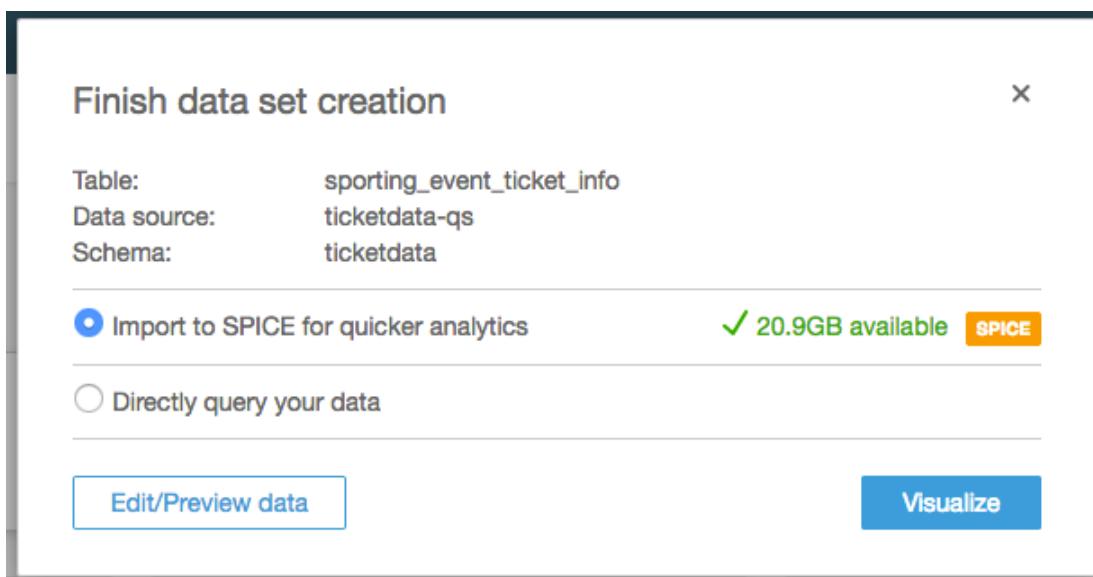
✓ Validated SSL is enabled Create data source

20. In the Database drop-down list, select the database name you created in the AWS Glue lab.

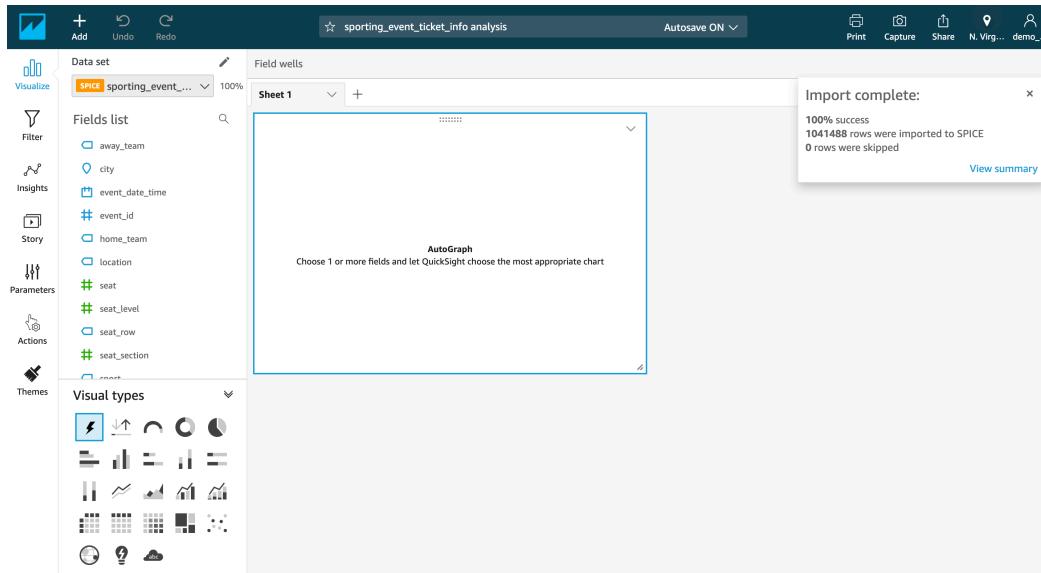
21. Choose the "sporting_event_ticket_info" table and click **Select**.



22. To finish data set creation, choose the option **Import to SPICE for quicker analytics** and click **Visualize**.



You will now be taken to the QuickSight Visualize interface where you can start building your dashboard.



Note: The SPICE dataset will take a few minutes to be built, but you can continue to create some charts on the underlying data.

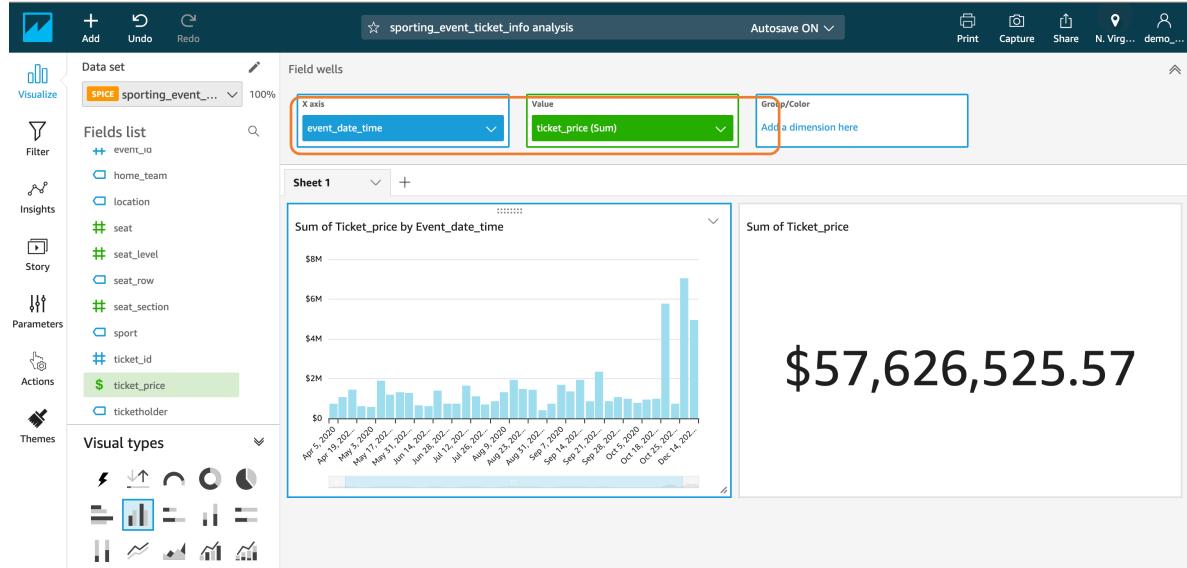
Create QuickSight Charts

In this section we will take you through some of the different chart types.

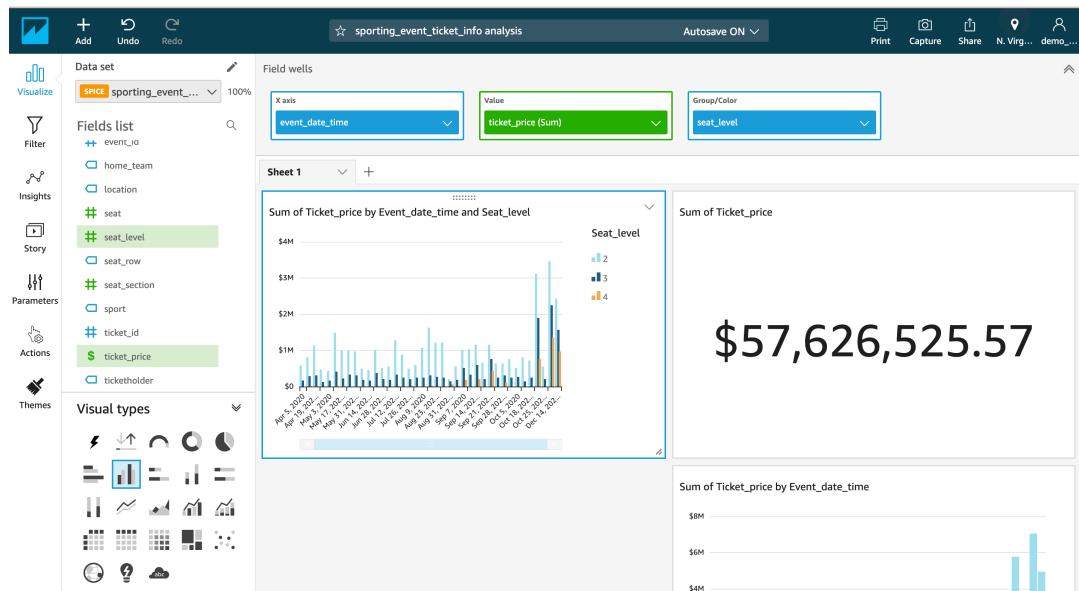
1. In the Fields list, click the "ticket_price" column to populate the chart.
2. Click the **expand icon** in corner of "ticket_price" field and select format as **currency** to show numbers in dollar amount.

3. You can add new visual and keep building your dashboard by clicking **Add button** at top left corner of screen.

In the **Visual types** area, choose the **Vertical bar chart** icon. This layout requires a value for the X-axis. In Fields list, select the "event_date_time" field and you should see the visualization update. For Y-axis, select "ticket_price" from the Field list.

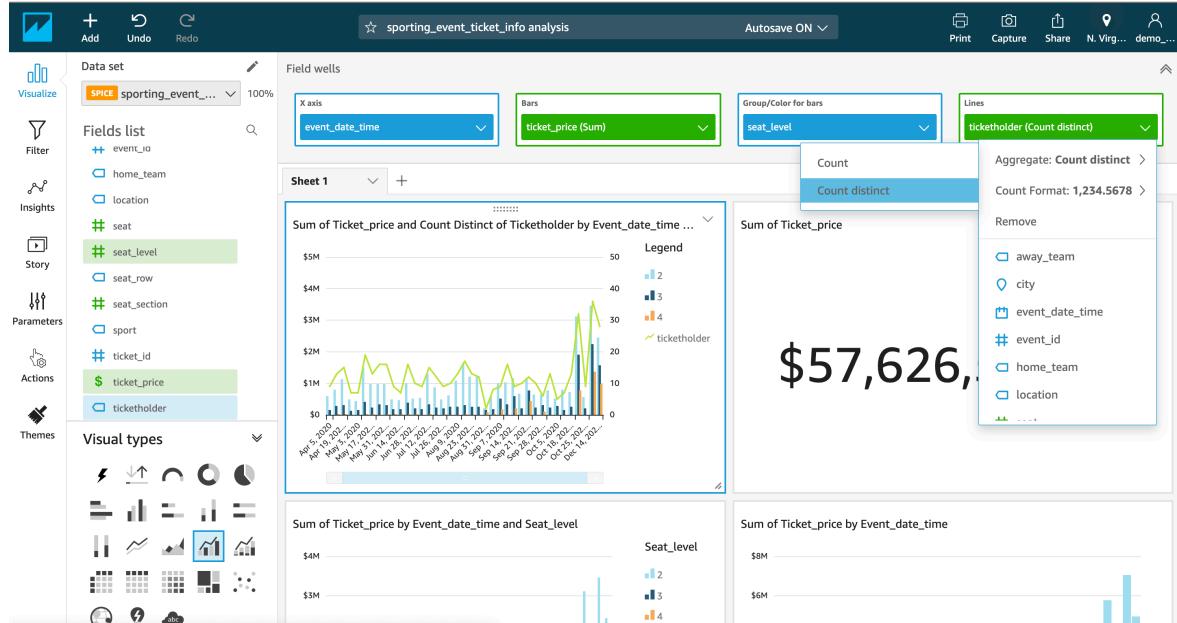


4. Add new Visual and you can drag and move other visuals to adjust space in dashboard. In the Fields list, click and drag the **seat_level** field to the **Group/Color** box in the **Field wells** pane. You can also use the slider below the x axis to fit all of the data.

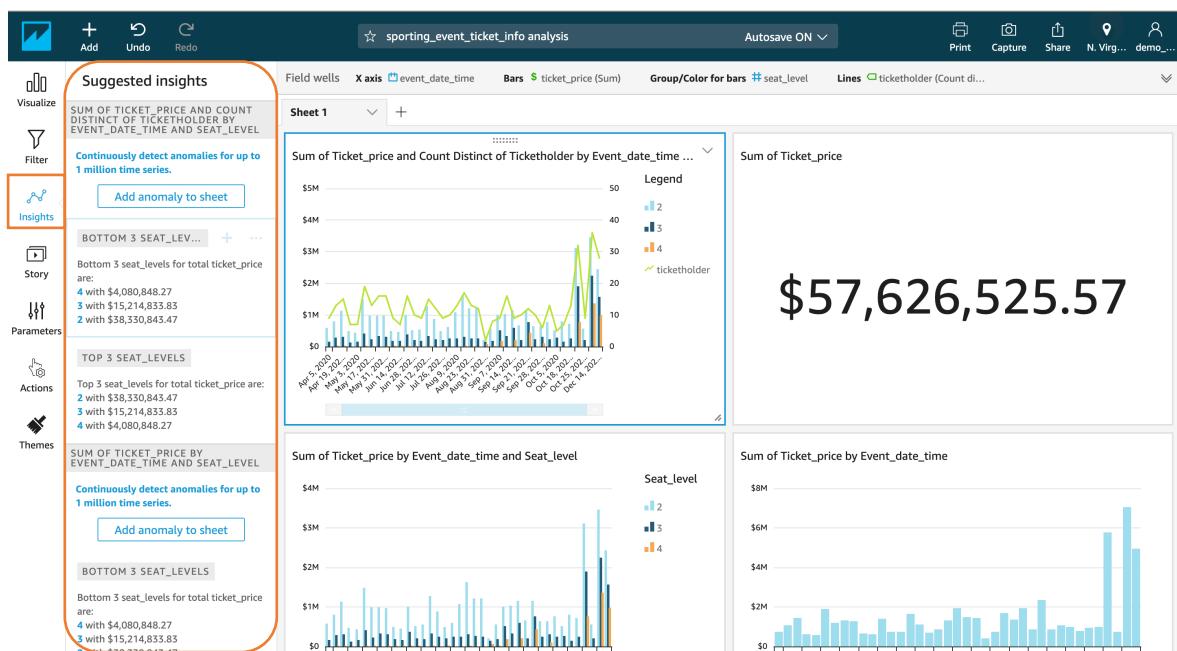


Let's build on this one step further by changing the chart type to "**Clustered bar combo chart**" and adding in the **ticketholder** field for the **Lines**.

5. In the Visual types area, choose the Clustered bar combo chart icon.
6. In the Fields list, click and drag the **ticketholder** field to the Lines box in the Field wells pane.
7. In the Field wells pane, click the Lines box and choose **Count Distinct** for Aggregate. You can then see the y-axis update on the right-hand side.



8. Click on **insight** icon on the left tabs section and explore insight information in simple English.

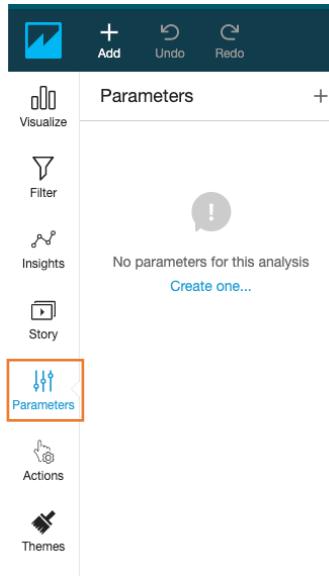


Feel free to experiment with other chart types and different fields to get a sense of the data.

Create QuickSight Parameters

In the next section we are going to create some parameters with controls for the dashboard, then assign these to a filter for all the visuals.

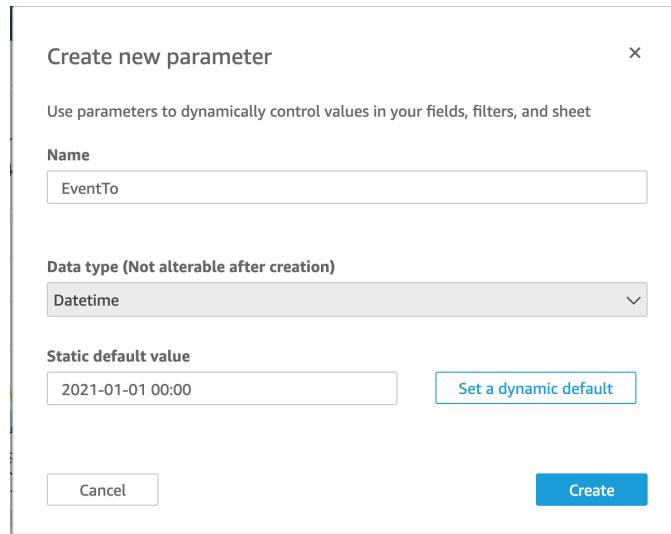
1. In the left navigation menu, select **Parameters**.



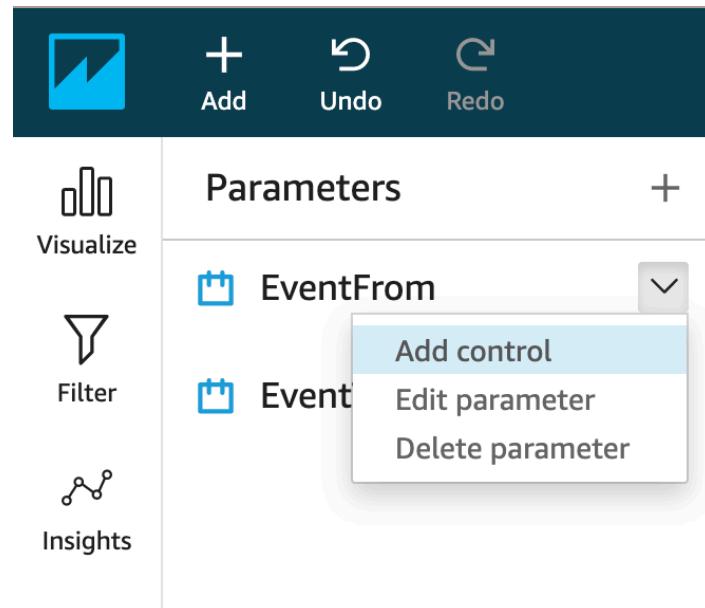
2. Click **Create one** to create a new parameter with a Name.
3. For Name, type **EventFrom**.
4. For Data type, choose **Datetime**.
5. For Default value, select the value from calendar as start date available in your graph for **event_date_time**. For example, **2020-01-01 00:00**.
6. Click **Create**, and then close the Parameter Added dialog box.

A screenshot of the 'Create new parameter' dialog box. It has a title bar 'Create new parameter' with a close button. Below it is a subtitle 'Use parameters to dynamically control values in your fields, filters, and sheet'. The 'Name' field contains 'EventFrom'. The 'Data type (Not alterable after creation)' dropdown is set to 'Datetime'. The 'Static default value' input field shows '2020-01-01 00:00'. There is a 'Set a dynamic default' button next to it. At the bottom are 'Cancel' and 'Create' buttons.

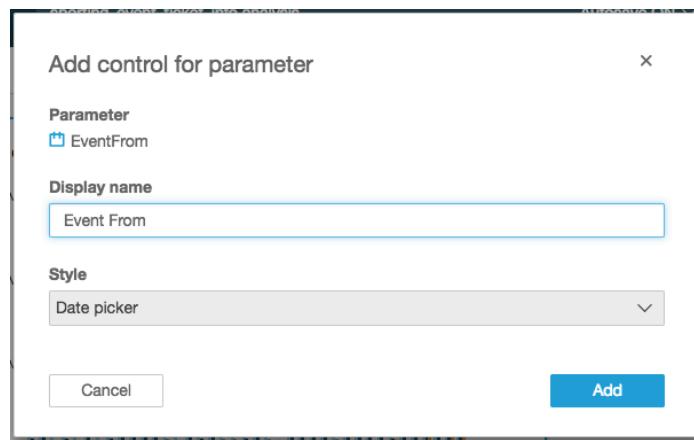
7. Create another parameter with the following attributes:
- Name: EventTo**
 - Data type: Datetime**
 - For Default value, select the value from calendar as end date available in your graph for **event_date_time**. For example, **2021-01-01 00:00**
 - Click **Create**



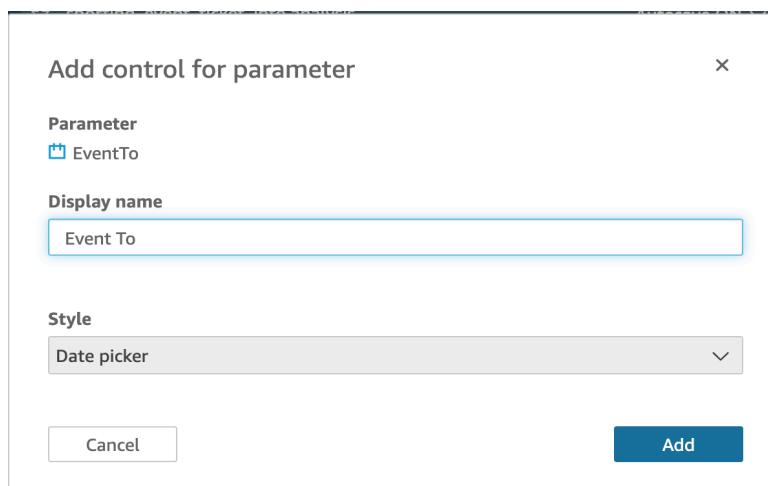
8. In next window, you can select any option to perform any operation with the parameter. Alternatively, you can click the drop-down menu for the **EventFrom** parameter and choose **Add control**.



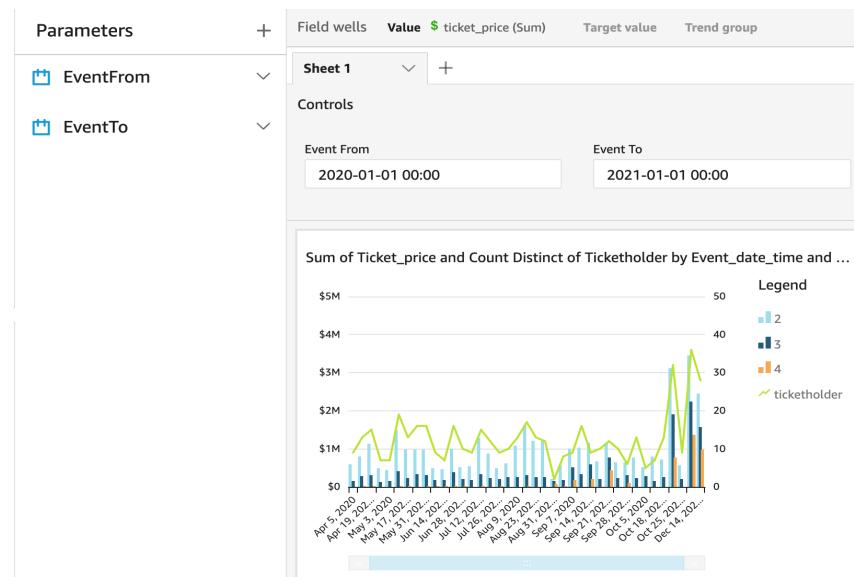
9. Click on **Control**
 10. For Display name, specify **Event From** and click **Add**.



11. Repeat the process to add a control for **EventTo** with display name **Event To**.



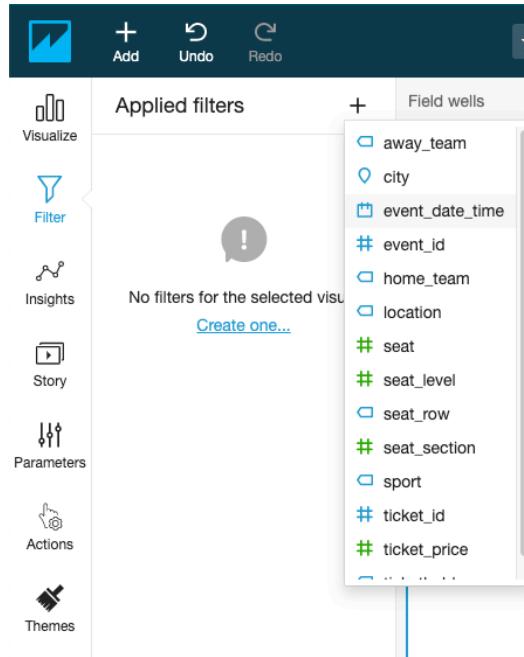
You should now be able to see and expand the Controls section above the chart.



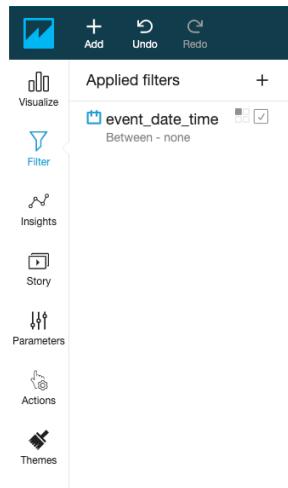
Create a QuickSight Filter

To complete the process, we will wire up a filter to these controls for all visuals.

1. In the left navigation menu, choose **Filter**.
2. Click the plus icon (+) to add a filter for the field "event_date_time".

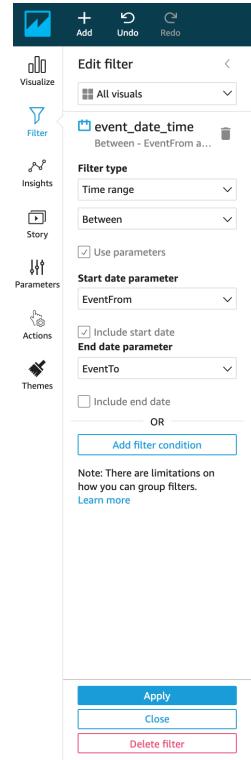


3. Click this filter to edit the properties.



4. Choose to make this filter apply to **All visuals**.
5. For Filter type, choose **Time range** and **Between**.
6. Select option **Use Parameter**.
7. For Start date parameter, choose **EventFrom**.

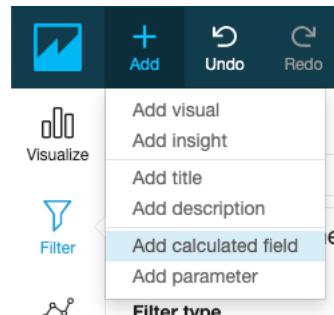
8. For **End date parameter**, choose **EventTo**.
9. Click **Apply**.



Add Calculated Fields

In the next section, you will learn, how to add calculated fields for "day of week" and "hour of day" to your dataset and a new scatter plot for these two dependent variables.

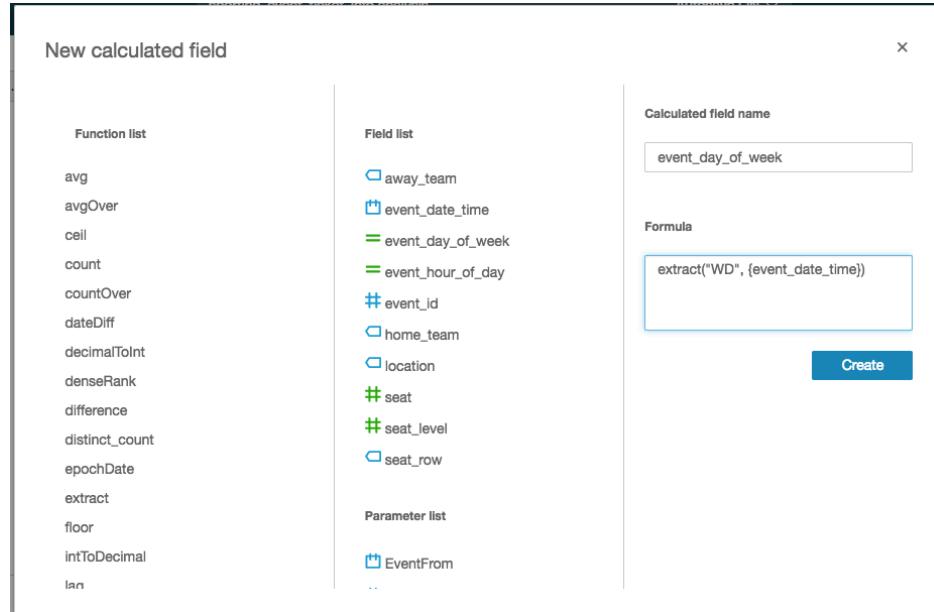
1. Click the Add button on the top left and select **Add a calculated field**.



2. For **Calculated field name** type "**event_day_of_week**".
3. For **Formula**, type `extract("WD",{event_date_time})`

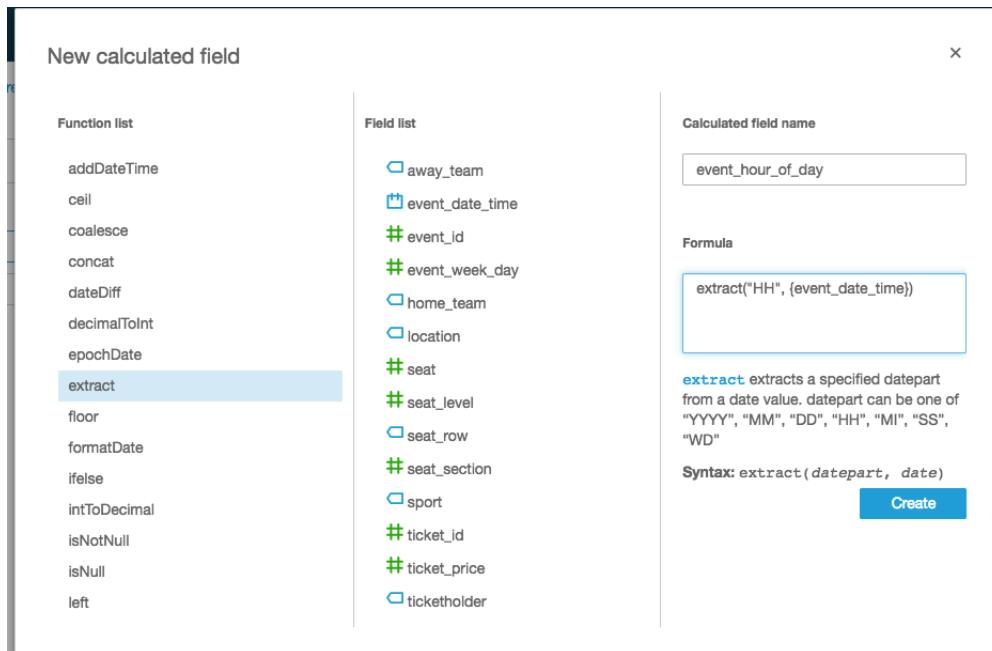
Note: extract returns a specified portion of a date value. Requesting a time-related portion of a date that doesn't contain time information returns 0. WD: This returns the day of the week as an integer, with Sunday as 1.

4. Click **Create**.

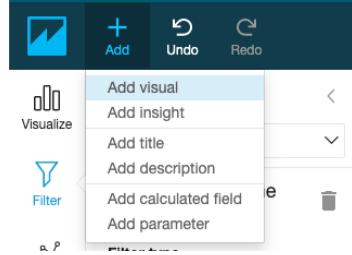


5. Add another calculated field with the following attributes:
- Calculated field name: "**event_hour_of_day**"
 - Formula: **extract("HH", {event_date_time})**

Note: HH: This returns the hour portion of the date.

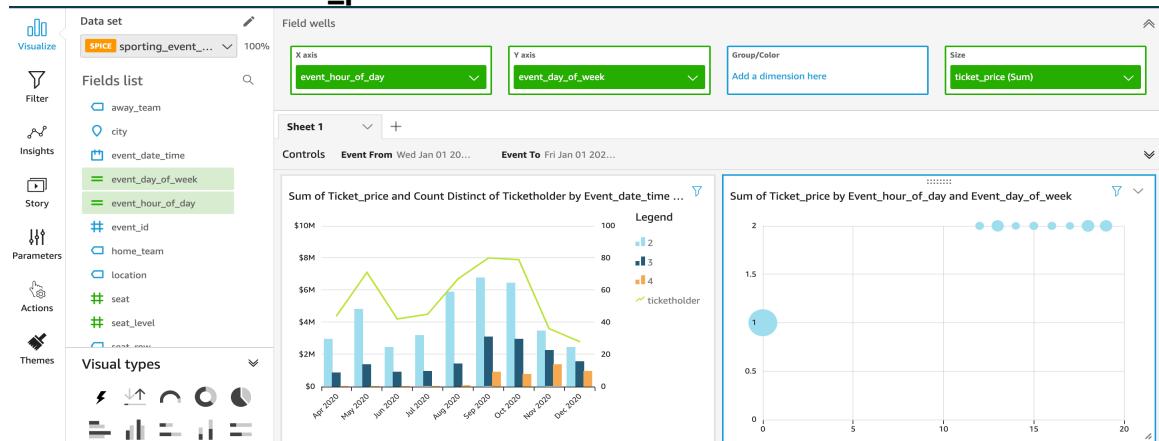


6. Click Add button in the top left and choose **Add visual**.

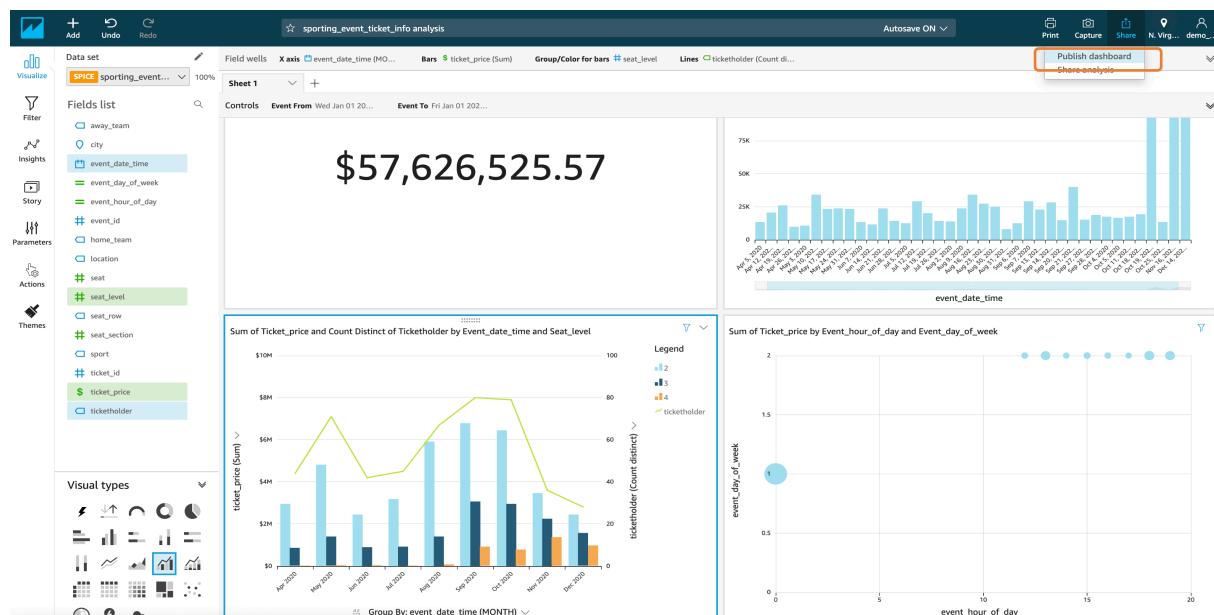


7. For field type, select the **scatter plot**.

8. In the Fields list, select and drag the following attributes to the Field wells pane to set the graph attributes:
 - a. **X-axis: "event_hour_of_day"**
 - b. **Y-axis: "event_day_of_week"**
 - c. **Size: "ticket_price"**



Since now you have completed your dashboard then you can **publish** it by clicking on top right corner of screen.



A *dashboard* is a read-only snapshot of an analysis that you can share with other Amazon QuickSight users for reporting purposes. In Dashboard other users can still play with visuals and data but that will not modify dataset.

You can share an analysis with one or more other users with whom you want to collaborate on creating visuals. Analysis provide other uses to write and modify data set.

Amazon QuickSight ML-Insights (Optional)

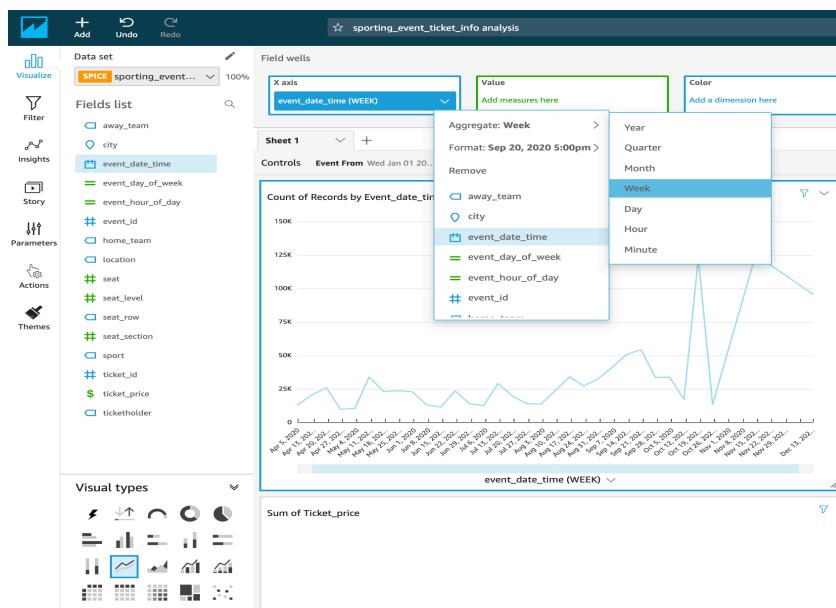
With Amazon QuickSight, you can add Machine Learning capabilities to your visuals, easily, with one click action. There are 3 types of Machine Learning Insights

- Narrative
- Anomaly Detection
- Forecasting

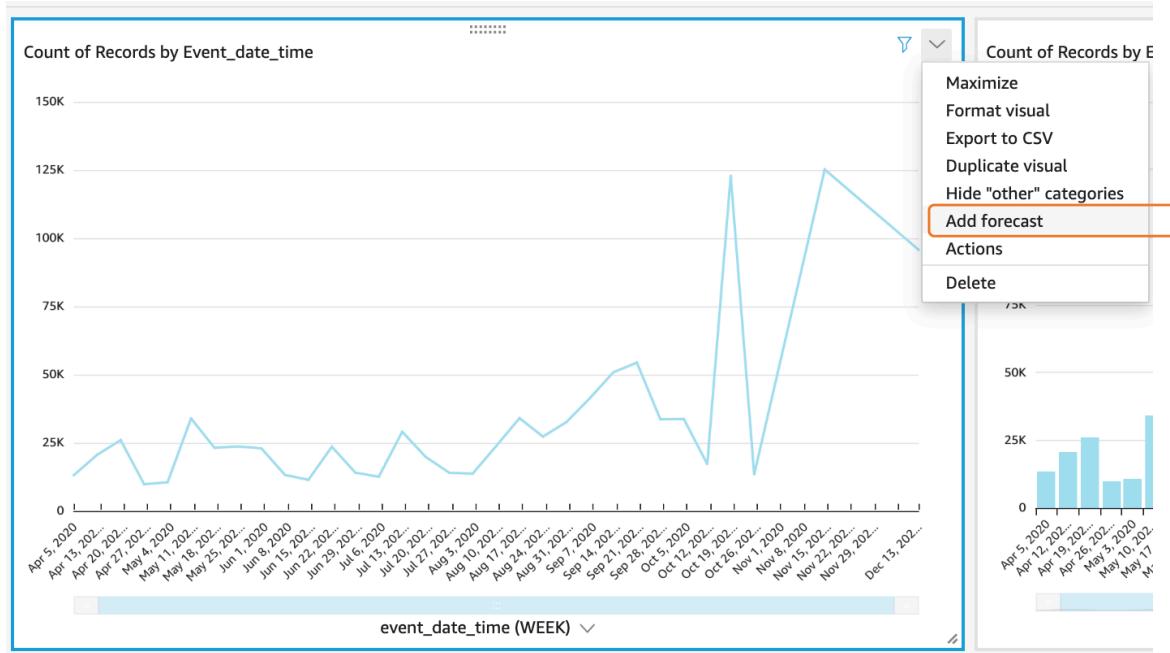
ML-Insights is only available to enterprise version of QuickSight. You will need to upgrade to Enterprise Edition before you start with the task. To upgrade your Amazon QuickSight Subscription from Standard Edition to Enterprise Edition please follow this guide <https://docs.aws.amazon.com/quicksight/latest/user/upgrading-subscription.html>

Let's see how we can add a bit of forecasting in our dashboard. Forecasting works with timeseries, which is better represented with a line graph. Let's first create a line graph.

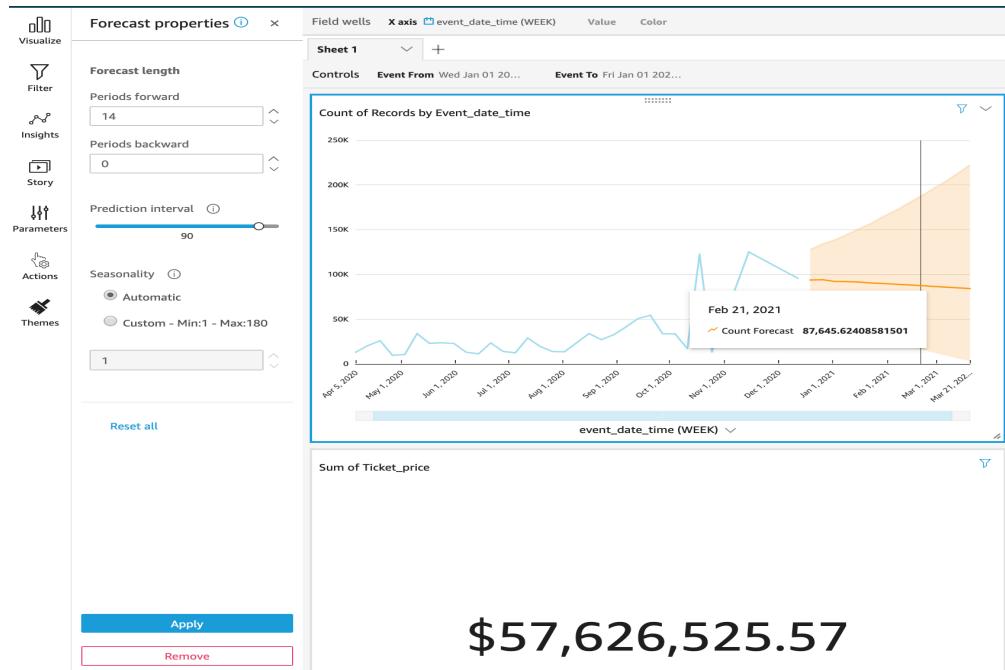
1. Click **add Visual at top left corner of screen**, and select **Line Chart** and add the **event_date_time** as the **x-axis** and **aggregate by week**. As shown in below screenshot



2. Add forecasting to the visual. To do that, click on the drop-down arrow on the top right corner of the visual, and then click **Add forecast**.



The visual will add forecast, you can hover over and explore forecasted data as shown below. Feel free to explore with the properties of the forecast algorithm.



Congratulations!! You have successfully completed this lab, Continue to Next section if you want to dive deep into Athena query access and cost

Athena Workgroups to Control Query Access and Costs (Optional)

Use workgroups to separate users, teams, applications, or workloads, to set limits on amount of data each query or the entire workgroup can process, and to track costs. Because workgroups act as resources, you can use resource-level identity-based policies to control access to a specific workgroup. You can also view query-related metrics in Amazon CloudWatch, control costs by configuring limits on the amount of data scanned, create thresholds, and trigger actions, such as Amazon SNS, when these thresholds are breached.

Workflow setup to separate workloads

For this lab, we will create two workgroups: “workgroupA” and “workgroupB”. Before creating the workgroups, you need to have users, appropriate IAM policies to assigned to each user and S3 buckets to store the query results. This has been created using Cloud Formation template for your convenience. It is recommended to go through the template for better understanding of pre-requisites. We will have two users: “business_analyst_user” and “workgroup_manager_user” created in IAM with different policies:

- The **business_analyst_user** will have access to **workgroupA** and query **sporting_event_info** table.
- The **workgroup_manager_user** will have access to both workgroups **workgroupA** and **workgroupB** for management purposes.

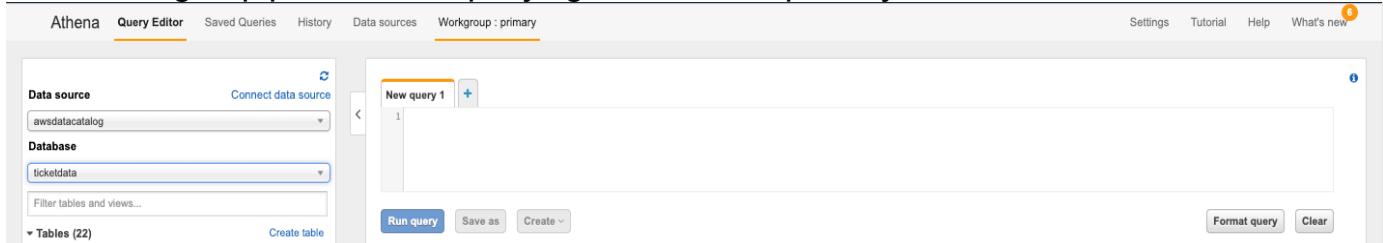
The resources have been already created as part of the DMS Student lab. You can click on the CloudFormation stack and navigate to “**Resources**” to understand the different resources created with “**DMSlab_student_CFN.json**” template. Navigate to **outputs** section to see the results of resources created with description.

Outputs (9)			
<input type="text"/> Search outputs			
Key	Value	Description	Export name
BucketName	dmslab-student-dmslabs3bucket-4a27jjap6c5t	S3 Bucket that was created	-
BusinessAnalystUser	dmslab-student-BusinessAnalystUser-878JWTT9AWCK	business_analyst_user for Workgroup A	-
BusinessAnalystUserPolicy	BusinessAnalystUserPolicy	User policy for Business Analyst User	-
DMSLabRoleS3	dmslab-student-DMSLabRoleS3-1VEPY3ZUJX9WB	The DMS service role	-
GlueLabRole	dmslab-student-GlueLabRole-Y0AJBNCP66ZI	The Glue service role	-
S3BucketWorkgroupA	dmslab-student-s3bucketworkgroupa-ldtj44qkwyle	S3 Bucket for storing workgroup A results	-
S3BucketWorkgroupB	dmslab-student-s3bucketworkgroupb-n2jrw40pfqcc	S3 bucket for storing workgroup B results	-
WorkgroupManagerUser	dmslab-student-WorkgroupManagerUser-KLF9GDANNNTVZ	workgroup_manager_user for access to Workgroup A and Workgroup B	-
WorkgroupManagerUserPolicy	WorkgroupManagerUserPolicy	User policy for Workgroup manager user	-

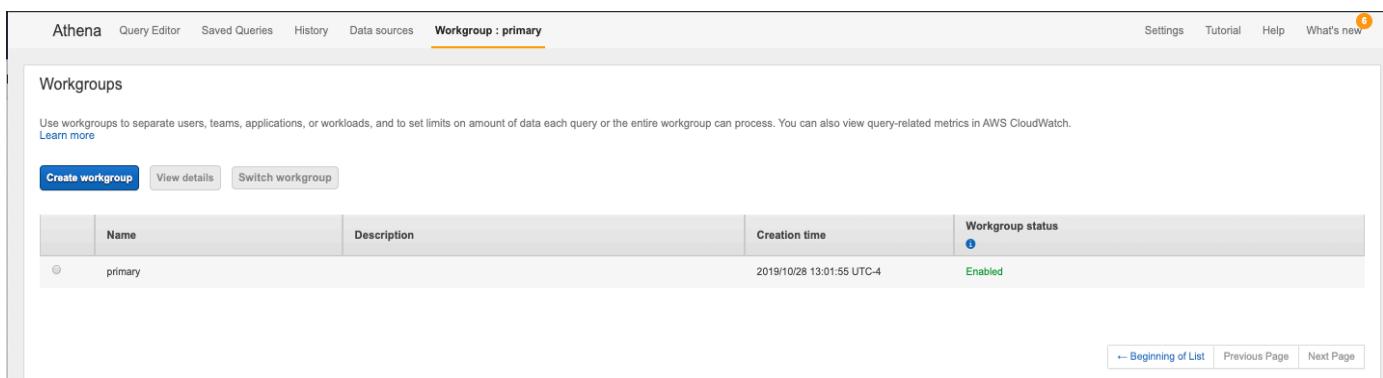
We will utilize the values from the outputs wherever required in the following steps.

Now we will create workgroups.

1. Navigate to **Athena Console** and click on “**Workgroup: primary**”. The default workgroup provided for querying in Athena is “primary”.



2. Click on “**Create workgroup**”



3. Provide the following:

- a. **Workgroup Name:** “workgroupA”
- b. **Description:** (optional)
 - i. “workgroupA for BusinessAnalystUser”
 - ii. “workgroupB for workgroup manager user”
- c. **Query result location:** Provide the query location, You can find S3 bucket name from Cloudformation output tab of student lab
 - i. For workgroupA, the s3 path would look something like: “s3://dmsslab-student-s3bucketworkgroupa-ldtj44qkwyle/”.
 - ii. For workgroupB, provide S3 path as: “s3://dmsslab-student-s3bucketworkgroupb-n2jrw40pfqcc/”.
- d. For “**Encrypt query results**”, leave as default i.e. unchecked. You can check this if you want your query results to be encrypted.
- e. Check the checkbox for “**Metrics: Publish query metrics to AWS CloudWatch**”

Workgroup name* workgroupA

Description workgroupA for BusinessAnalystUser

Use up to 1024 characters.

Query result location and Encryption

Query result location s3://dmslab-student-s3bucketworkgroupa-ldtj44 Select Enter a path to an S3 bucket or prefix.

Encrypt query results Encrypt results stored in S3

Metrics

Metrics Publish query metrics to AWS CloudWatch [?](#)

Settings

Override client-side settings [?](#)

Requester pays S3 buckets Enable queries on requester pays buckets in Amazon S3 [?](#)

Tags

A tag is a label that you assign to an Athena workgroup resource. It consists of a key and a value. Use tags to categorize workgroups by purpose, owner, or environment. You can also use tag specific values for a tag key. Use [best practices](#) and create a consistent set of tags. Do not use duplicate tag keys the same workgroup. [Learn more](#)

Key	name	Value (Optional)	workgroupA	x
Use 1 - 128 characters. (A-Z,a-z,0-9,_,:,;=/+,_,@)		Use up to 256 characters. (A-Z,a-z,0-9,_,:/=+,_,@)		

4. Provide the following:

- a. Optionally, you can click on **Override client-side settings**. This will override the client-side settings and keep the defaults for query execution and storing results.
- b. **Tag** your workgroup to analyze later with CloudWatch or perform any analytics on query execution and results.
 - i. For **workgroupA**: provide **key:"name", value:"workgroupA"**
 - ii. For **workgroupB**: Provide **key:"name", value:"workgroupB"**
- c. For "**Requester Pays S3 buckets**", keep as **default**. This is **Optional**. Choose **Enable queries on Requester Pays buckets in Amazon S3** if workgroup users will run queries on data stored in Amazon S3 buckets that are configured as Requester Pays. The account of the user running the query is charged for applicable data access and data transfer fees associated with the query.

5. Click on **create workgroup**

6. Follow the above procedure to create **workgroupB**.

Explore the features of workgroups

- From the **Outputs** tab of DMS student lab, Note down user name **BusinessAnalystUser** and bucket name **S3BucketWorkgroupA** and save it.

Outputs (9)			
Key	Value	Description	Export name
BucketName	dmslab-student-dmslabs3bucket-4a27jjap6c5t	S3 Bucket that was created	-
BusinessAnalystUser	dmslab-student-BusinessAnalystUser-878JWTT9AWCK	business_analyst_user for Workgroup A	-
BusinessAnalystUserPolicy	BusinessAnalystUserPolicy	User policy for Business Analyst User	-
DMSLabRoleS3	dmslab-student-DMSLabRoleS3-1VEPY3ZUJX9WB	The DMS service role	-
GlueLabRole	dmslab-student-GlueLabRole-YOAJBNCP66ZI	The Glue service role	-
S3BucketWorkgroupA	dmslab-student-s3bucketworkgroupa-ldtj44qkwyle	S3 Bucket for storing workgroup A results	-
S3BucketWorkgroupB	dmslab-student-s3bucketworkgroupb-n2jrw40pfqcc	S3 bucket for storing workgroup B results	-
WorkgroupManagerUser	dmslab-student-WorkgroupManagerUser-KLF9GDANNTVZ	workgroup_manager_user for access to Workgroup A and Workgroup B	-
WorkgroupManagerUserPolicy	WorkgroupManagerUserPolicy	User policy for Workgroup manager user	-

- Note down 12 digit AWS account id . Follow steps here to find out account id - <https://www.apn-portal.com/knowledgebase/articles/FAQ/Where-Can-I-Find-My-AWS-Account-ID>
- Next, Open [AWS console log-in](#) different browser, select **IAM user** and login with following credential:
 - AccountID:** <your-account-name-or-alias>
 - IAM User name:** <value copied for **BusinessAnalystUser**>
 - Password:** **master123**
- From new **BusinessAnalystUser** user, Navigate to Athena Console . You will notice that you can see your workgroup designated as "workgroupA" and you can also view table: **sporting_event_info** as shown below:

The screenshot shows the AWS Athena console interface. At the top, there's a navigation bar with links for Services, Resource Groups, Athena, S3, AWS Glue, RDS, and other options like Lakeformation, N. Virginia, Support, Settings, Tutorial, Help, and What's new. The main area has tabs for 'Athena' (which is selected) and 'Query Editor'. In the 'Athena' tab, there are sections for 'Data source' (set to 'AwsDataCatalog'), 'Database' (set to 'ticketdata'), and 'Tables' (listing 'parquet_sport_location', 'parquet_sport_team', 'parquet_sporting_event', and 'Views (1)' which includes 'sporting_event_info'). The 'Query Editor' tab is active, showing a query editor with a single query: 'SELECT * FROM "ticketdata"."sporting_event_info" limit 10;'. Below the query editor, there's a 'Run query' button, a 'Save as' dropdown, and a 'Create' dropdown. A note says 'Use Ctrl + Enter to run query, Ctrl + Space to autocomplete'. The bottom section is labeled 'Results' and currently shows an empty table.

If your workgroup is other than **workgroupA**, click on Workgroup:

The screenshot shows the AWS Athena Query Editor interface. At the top, there are tabs for Services, Resource Groups, Athena, S3, AWS Glue, and RDS. Below the tabs, the 'Workgroup' dropdown is set to 'workgroupA'. The main area has two tabs: 'New query 1' and 'New query 2'. 'New query 2' is active, containing the SQL query: 'SELECT * FROM "ticketdata"."sporting_event_info" limit 10;'. Below the query editor are sections for 'Data source', 'Database', and 'Tables (3)'. The 'Tables' section lists 'parquet_sport_location', 'parquet_sport_team', and 'parquet_sporting_event'. There is also a 'Views (1)' section with 'sporting_event_info'. On the right side, there are buttons for 'Run query', 'Save as', and 'Create', along with 'Format query' and 'Clear' options.

Select **workgroupA** from the workgroup list and then click on **Switch Workgroup**.

Workgroups

Use workgroups to separate users, teams, applications, or workloads, and to set limits on amount of data each query or the entire workgroup can process. You can also view query-related metrics in AWS CloudWatch. [Learn more](#)

The screenshot shows the AWS Workgroups console. At the top, there are buttons for 'Create workgroup', 'View details', and 'Switch workgroup'. The 'Switch workgroup' button is highlighted with an orange border. Below is a table listing workgroups:

	Name	Description	Creation time	Workgroup status
●	workgroupB	workgroupB for WorkgroupManagerUser	2020/03/13 20:09:48 UTC-4	Enabled
●	workgroupA	workgroupA for BusinessAnalystUser	2020/03/13 20:02:31 UTC-4	Enabled
●	primary		2020/03/13 19:35:43 UTC-4	Enabled

5. If you see that your bucket is not setup with Athena to store the query results, as shown below, then proceed to setup the bucket.

The screenshot shows the AWS Athena Query Editor interface. At the top, there are tabs for Services, Resource Groups, Athena, S3, AWS Glue, and RDS. Below the tabs, the 'Workgroup' dropdown is set to 'primary'. The main area has two tabs: 'New query 1' and 'New query 2'. 'New query 2' is active, containing the SQL query: 'SELECT * FROM "ticketdata"."sporting_event_info" limit 10;'. A callout box highlights a message: 'Before you run your first query, you need to [set up a query result location in Amazon S3](#)'. Below the message, there are buttons for 'Run query', 'Save as', and 'Create'.

6. Setup the S3 bucket for storing the query results. Click on **Settings**.

The screenshot shows the AWS Athena Query Editor interface. On the left, the sidebar displays the data source (AwsDataCatalog), database (ticketdata), and tables (parquet_sport_location, parquet_sport_team, parquet_sporting_event) and views (sporting_event_info). The main area shows a query editor with a single query: "SELECT * FROM \"ticketdata\".\"sporting_event_info\" limit 10;". Below the query are buttons for Run query, Save as, and Create. A status bar at the bottom indicates "(Run time: 2.82 seconds, Data scanned: 12.67 KB)".

Provide the S3 bucket location for workgroupA, copied and saved from the Output tab of cloud formation template, as shown below. Then, click on **Save**.

The screenshot shows the Settings dialog box for workgroupA. It contains fields for Query result location (set to s3://dmslab-student-s3bucketworkgroupa-ltdj44qkwyle/), Encrypt query results (unchecked), and Autocomplete (unchecked). At the bottom are Cancel and Save buttons.

- Back to Athena Query Editor, click on the three dots against **sporting_event_info** view and then click on **Preview**. You will be able to see query results. This shows that you as “business_analyst_user” has access to query the view **sporting_event_info** and store the query results in S3 bucket designated for workgroupA.

The screenshot shows the Athena Query Editor with the sporting_event_info view selected in the sidebar. The main area displays the query "SELECT * FROM \"ticketdata\".\"sporting_event_info\" limit 10;" and its execution results. The results table is highlighted with an orange border. The table contains 10 rows of data, including columns like event_id, sport, event_date_time, home_team, away_team, location, and city.

	event_id	sport	event_date_time	home_team	away_team	location	city
1	4491	baseball	2020-07-19 00:00:00.000	Seattle Mariners	New York Mets	Safeco Field	Seattle Washington
2	4581	baseball	2020-09-20 00:00:00.000	Seattle Mariners	Boston Red Sox	Safeco Field	Seattle Washington
3	4191	baseball	2020-07-05 00:00:00.000	Houston Astros	Texas Rangers	Minute Maid Park	Houston Texas
4	4451	baseball	2020-06-14 00:00:00.000	Seattle Mariners	Philadelphia Phillies	Safeco Field	Seattle Washington
5	4481	baseball	2020-07-05 00:00:00.000	Seattle Mariners	Toronto Blue Jays	Safeco Field	Seattle Washington
6	4611	baseball	2020-10-11 00:00:00.000	Seattle Mariners	Los Angeles Angels	Safeco Field	Seattle Washington
7	5661	baseball	2020-07-26 00:00:00.000	San Diego Padres	Kansas City Royals	Petco Park	San Diego California
8	5591	baseball	2020-05-31 00:00:00.000	San Diego Padres	Texas Rangers	Petco Park	San Diego California
9	9151	football	2020-09-21 19:00:00.000	San Diego Chargers	Denver Broncos	Qualcomm Stadium	San Diego, California
10	4071	baseball	2020-04-12 00:00:00.000	Houston Astros	Tampa Bay Rays	Minute Maid Park	Houston Texas

- Logged in as "business_analyst_user", click on **workgroup** and try switching to other workgroups which this user does not have access to. Select **workgroupB** and then click on **switch workgroup**.

Workgroups

Use workgroups to separate users, teams, applications, or workloads, and to set limits on amount of data each query or the entire workgroup can process. You can also view query-related metrics in AWS CloudWatch. [Learn more](#)

	Name	Description	Creation time	Workgroup status
<input checked="" type="radio"/>	workgroupB	workgroupB for WorkgroupManagerUser	2020/03/13 20:09:48 UTC-4	Enabled
<input type="radio"/>	workgroupA	workgroupA for BusinessAnalystUser	2020/03/13 20:02:31 UTC-4	Enabled
<input type="radio"/>	primary		2020/03/13 19:35:43 UTC-4	Enabled

- If you try running the query, you will get the error "Access Denied" as shown below:

The screenshot shows the AWS Athena Query Editor interface. On the left, there's a sidebar with 'Data source' set to 'awsdatasatalog' and 'Database' set to 'ticketdata'. Below these are sections for 'Tables (3)' and 'Views (1)'. In the main area, there's a query editor window with a red box highlighting an error message: 'Your query has the following error(s): User: arn:aws:iam::665953140268:user/lakeformation-BusinessAnalystUser-7H32WD4CWS6 is not authorized to perform: athena:StartQueryExecution on resource: arn:aws:athena:us-east-1:665953140268:workgroup/workgroupB (Service: AmazonAthena; Status Code: 400; Error Code: AccessDeniedException; Request ID: 40b3397b-f49b-4d1c-b44c-cdcaaf47e977)'.

This means that we have achieved the user segregation for different workgroups as defined by the IAM policy and attached to that user. Any query executed and its results within a particular workgroup will be isolated to that workgroup.

- To view the query results, navigate to "workgroup", select the workgroup and click on "View Details".

Workgroups

Use workgroups to separate users, teams, applications, or workloads, and to set limits on amount of data each query or the entire workgroup can process. You can also view query-related metrics in AWS CloudWatch. [Learn more](#)

	Name	Description	Creation time	Workgroup status
<input checked="" type="radio"/>	workgroupB	workgroupB for WorkgroupManagerUser	2020/03/13 20:09:48 UTC-4	Enabled
<input checked="" type="radio"/>	workgroupA	workgroupA for BusinessAnalystUser	2020/03/13 20:02:31 UTC-4	Enabled
<input type="radio"/>	primary		2020/03/13 19:35:43 UTC-4	Enabled

11. You will be able to see the details, as shown below. Navigate to S3 bucket by clicking on the link and see the query results stored inside the "Unsaved" folder within the **workgroupA** bucket.

12. Now, login as `workgroup_manager_user`.

- Account ID or Alias: <you-account-id-or-alias>
- IAM User Name: <Copy the IAM User Name from cloud formation outputs tab> (for e.g: in this lab: dmslab-student-WorkgroupManagerUser-KLF9GDANNTVZ)
- Password: master123

This user has access to `workgroupA` and `workgroupB` for management purposes. Switch the workgroups to `workgroupA`, `workgroupB` and primary and you will not be able to access the primary workgroup because this user **does not have access to “primary” workgroup**.

	Name	Description	Creation time	Workgroup status
●	workgroupB	workgroupB for WorkgroupManagerUser	2020/03/13 20:09:48 UTC-4	Enabled
●	workgroupA	workgroupA for BusinessAnalystUser	2020/03/13 20:02:31 UTC-4	Enabled
●	primary		2020/03/13 19:35:43 UTC-4	Enabled

Also note that this user does not have access to any tables or cannot run any queries. This is how we can isolate the responsibilities of different users within different workgroups by defining policies and attaching that to the user.

At any point of time, **you can edit, delete and disable your workgroups** as shown:

Select the workgroup and click on “**View Details**”.

Workgroups

Use workgroups to separate users, teams, applications, or workloads, and to set limits on amount of data each query or the entire workgroup can process. You can also view query-related metrics in AWS CloudWatch. [Learn more](#)

	Name	Description	Creation time	Workgroup status
●	workgroupB	workgroupB for WorkgroupManagerUser	2020/03/13 20:09:48 UTC-4	Enabled
●	workgroupA	workgroupA for BusinessAnalystUser	2020/03/13 20:02:31 UTC-4	Enabled
●	primary		2020/03/13 19:35:43 UTC-4	Enabled

Click on “**Edit Workgroup**” to make changes, “**Delete workgroup**” to delete the entire workgroup and “**Disable workgroup**” to disable the workgroup and disable any queries to be run within that workgroup.

Workgroup: workgroupA

[Edit workgroup](#) [Delete workgroup](#) [Disable workgroup](#) [Enable workgroup](#)

[Overview](#) [Metrics](#) [Data usage controls](#) [Tags](#)

To grant access to the workgroup, [create an IAM policy](#) and attach it to a user, group or role. [Learn more](#)

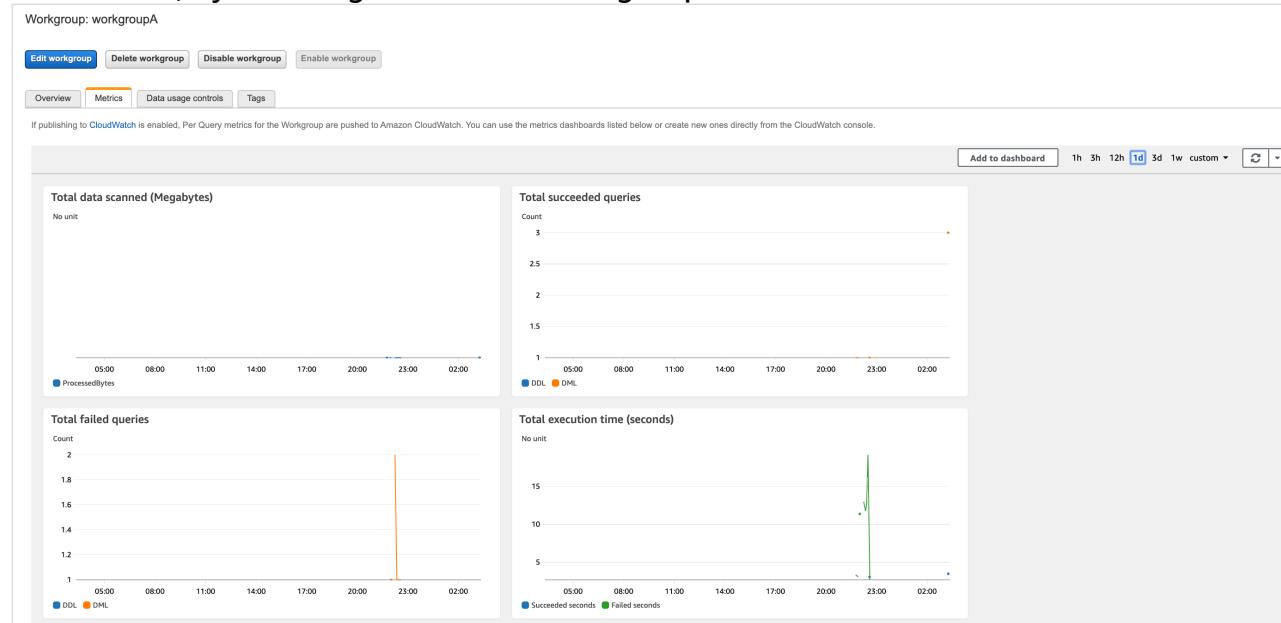
Description	Not defined
Query result location	s3://dmslab-student-s3bucketworkgroupa-ldj44qkwyle/ Edit
Amazon CloudWatch Metrics	Enabled
Encrypt query results	Not defined
Workgroup status	Enabled
Workgroup ARN	arn:aws:athena:us-east-1:678691952726:workgroup/workgroupA Edit
Bytes scanned cut off per query	Not defined
Override client-side settings	Disabled
Queries with requester pays buckets	Disabled

Please Note: For lab purpose, we are attaching policies directly to users. For Best practices, we recommend creating separate groups in IAM for different workgroups and then attaching policies for different workgroups to their respective groups in IAM.

Managing Query Usage and Cost

****Please Note** that the following section of this lab is carried out under **admin** account and not the **BusinessAnalystUser** and **WorkgroupManagerUser**, so please login to your account with admin credentials**

Once you **enable the CloudWatch metrics** for your workgroups, you will be able to see **Metrics**, by selecting the desired workgroup and click on **Metrics** as shown:



Choose the **metrics interval** that Athena should use to fetch the query metrics from CloudWatch, or choose the **refresh** icon to refresh the displayed metrics.



Let's setup data usage controls which means setting up the threshold for the amount of data scanned. There are two types of data usage controls: **per-query** and **per-workgroup**.

Per-query data usage control will check the total amount of data scanned by per query within the workgroup and if the amount exceeds the threshold, the query will be cancelled automatically. Let's setup **per-query data usage for “primary workgroup”**.

1. From Athena console, click on **Workgroup** and select **primary**. Click on **View Details**

Workgroups

Use workgroups to separate users, teams, applications, or workloads, and to set limits on amount of data each query or the entire workgroup can process. You can also view query-related metrics in AWS CloudWatch. [Learn more](#)

[Create workgroup](#) [View details](#) [Switch workgroup](#)

Name	Description	Creation time	Workgroup status
workgroupB	workgroupB for WorkgroupManagerUser	2020/03/13 20:09:48 UTC-4	Enabled
workgroupA	workgroupA for BusinessAnalystUser	2020/03/13 20:02:31 UTC-4	Enabled
primary		2020/03/13 19:35:43 UTC-4	Enabled

2. Click on **Data usage controls**. In **Per query data usage control**, the default minimum limit is 10MB per query. We will select the default value- 10MB. Also, note the default “**Action**” for per query data usage control. **If the query exceeds the limit, it will be cancelled**.
3. Click **Update**
4. The per-query threshold has been set.

Workgroup: primary

[Edit workgroup](#) [Delete workgroup](#) [Disable workgroup](#) [Enable workgroup](#)

[Overview](#) [Metrics](#) [Data usage controls](#) [Tags](#)

Per query data usage control

Sets the limit for the maximum amount of data a query is allowed to scan. You can set only one per query limit for a workgroup. The limit applies to all queries in the workgroup. [Learn more](#)

Data limits	10	Megabytes MB
-------------	----	--------------

Minimum Limit 10MB per query.

Action If the query exceeds the limit, it will be cancelled.

[Delete](#) [Update](#)

5. Navigate to query editor on Athena console. Run the following query:

```
SELECT * FROM "ticketdata"."cdc_sporting_event_ticket"
```
6. This query scans 200 MB of data, but since we have set the threshold as 10MB, this query execution will be cancelled, as shown:

The screenshot shows the AWS Athena Query Editor interface. On the left, there are dropdown menus for 'Data source' (awsdatacatalog), 'Database' (ticketdata), and 'Tables (24)' (including bookmark_parquet_ticket_purchase_history, cdc_sporting_event_ticket, and vnde_ticket_purchase_hist). In the main area, a query window contains the command: `1: SELECT * FROM "ticketdata"."cdc_sporting_event_ticket"`. Below the query window, status information shows '(Run time: 2.51 seconds, Data scanned: 10.49 MB)'. A red alert box at the top right displays the message: 'Query cancelled! : Bytes scanned limit was exceeded'. At the bottom of the interface, there are buttons for 'Run query', 'Save as', and 'Create'.

For **per-workgroup data usage control**, you can configure the maximum amount of data scanned by all queries in the workgroup during a specific period. This is useful when you have few analytics reports to run, where you probably have a good idea of how long the process should take and the total amount of data that queries scan during this time. You only have a few reports to run, so you can expect them to run in a few minutes, only scanning a few megabytes of data.

1. Login as **Admin** to the account. On Athena console, click on **Workgroup** and Select **workgroupA**. Click on **View Details**.

Workgroups

Use workgroups to separate users, teams, applications, or workloads, and to set limits on amount of data each query or the entire workgroup can process. You can also view query-related metrics in AWS CloudWatch. [Learn more](#)

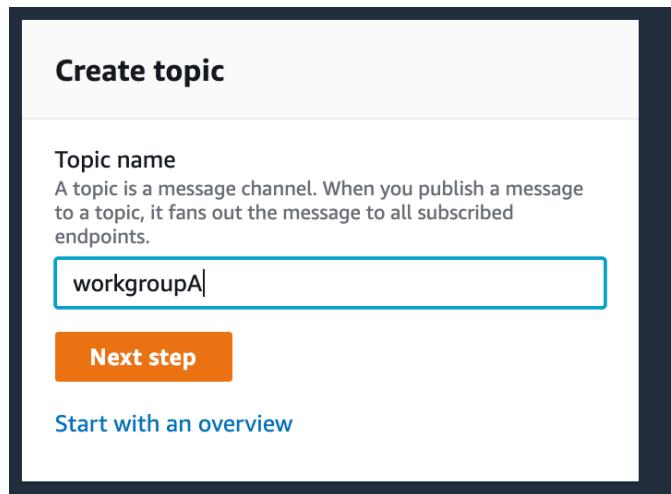
The screenshot shows the AWS Workgroups page. At the top, there are buttons for 'Create workgroup', 'View details' (which is highlighted with an orange border), and 'Switch workgroup'. Below is a table listing three workgroups:

Name	Description	Creation time	Workgroup status
workgroupB	workgroupB for WorkgroupManagerUser	2020/03/13 20:09:48 UTC-4	Enabled
workgroupA	workgroupA for BusinessAnalystUser	2020/03/13 20:02:31 UTC-4	Enabled
primary		2020/03/13 19:35:43 UTC-4	Enabled

2. Click on **Data usage Controls** and scroll down to section **Workgroup data usage controls**. Click on **Create workgroup data usage control**

3. The select query on **sporting_event_info** returns more than 10KB of data. For this lab, we have only this table to query from. So, let's set the threshold accordingly.

- Set **Data Limits** to **10 KBs**
- Set **Time period** to **1 minute**
- Set **Action** as "**Send a notification to**". Here, click on **Create SNS Topic**.
 - This will take you to **SNS Console**. Provide **Topic Name** as **workgroupA**.



- Click on **Next Step**
- Click on **Create Subscription**. We will subscribe to this topic with **email address**. Whenever the threshold is breached, we will get an email notification to the email address which is our subscriber.

- iv. In **Create Subscription**, select **Protocol** as **Email**. In **Endpoint**, Provide **email address**, then click on **Create subscription**.

- v. Verify your email for subscription to be validated.
 vi. Back to WorkgroupA workgroup data usage control, for **Action**, select **workgroupA** for the **SNS topic**. Click on **Create**.

Create workgroup data usage control

Sets the limit for the maximum amount of data queries running in this workgroup are allowed to scan within a specific period. The limit applies to all queries in the workgroup. You can set multiple limits per workgroup, and trigger different actions for each of them. Limits are implemented as [AWS CloudWatch alarms](#), and you can trigger [actions](#) when those alarms are breached. [Learn more](#)

Data limits	10	Kilobytes
Time period	1 minute	
Action	<input checked="" type="checkbox"/> Send a notification to workgroupA Create SNS topic	
		Cancel Create

vii. Once created, this control will be listed like this:

Workgroup: workgroupA																
Edit workgroup Delete workgroup Disable workgroup Enable workgroup																
Overview Metrics Data usage controls Tags																
Per query data usage control																
Sets the limit for the maximum amount of data a query is allowed to scan. You can set only one per query limit for a workgroup. The limit applies to all queries in the workgroup. Learn more																
<table border="1"> <tr> <td>Data limits</td> <td>10</td> <td>Megabytes MB</td> </tr> <tr> <td colspan="3">Minimum Limit 10MB per query.</td> </tr> <tr> <td>Action</td> <td colspan="2">If the query exceeds the limit, it will be cancelled.</td> </tr> <tr> <td colspan="3"> Delete Update </td> </tr> </table>					Data limits	10	Megabytes MB	Minimum Limit 10MB per query.			Action	If the query exceeds the limit, it will be cancelled.		Delete Update		
Data limits	10	Megabytes MB														
Minimum Limit 10MB per query.																
Action	If the query exceeds the limit, it will be cancelled.															
Delete Update																
Workgroup data usage controls																
Sets the limit for the maximum amount of data queries running in this workgroup are allowed to scan within a specific period. The limit applies to all queries in the workgroup. You can set multiple limits per workgroup, and trigger different actions for each of them. Limits are implemented as AWS CloudWatch alarms , and you can trigger actions when those alarms are breached. Learn more																
<table border="1"> <tr> <th>Create</th> <th>Delete</th> </tr> <tr> <th></th> <th>Data limits</th> <th>Time period</th> <th>Action</th> <th></th> </tr> <tr> <td>10.24 KB</td> <td>1 minute</td> <td></td> <td>Send notification to topic : arn:aws:sns:us-east-1:865953140268:workgroupA</td> <td></td> </tr> </table>					Create	Delete		Data limits	Time period	Action		10.24 KB	1 minute		Send notification to topic : arn:aws:sns:us-east-1:865953140268:workgroupA	
Create	Delete															
	Data limits	Time period	Action													
10.24 KB	1 minute		Send notification to topic : arn:aws:sns:us-east-1:865953140268:workgroupA													

4. Back to **Athena Query Editor**, run the following query, by logging in as **Business Analyst User** to the console and selecting **Workgroup: workgroupA**:

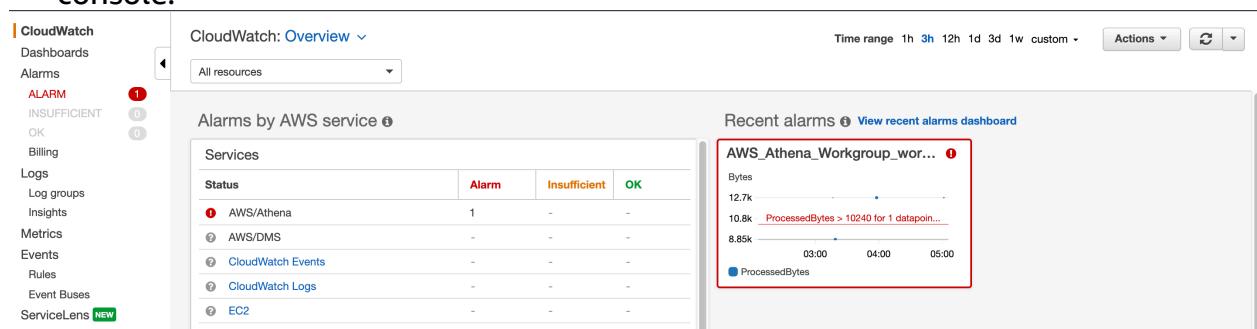
```
SELECT * FROM "ticketdata"."sporting_event_info";
```

5. You will receive an **email notification from AWS Notifications** stating that workgroup data usage threshold has been breached, which will look something like this:

ALARM: "AWS_Athena_Workgroup_workgroupA_c0ff968d-32fe-4c37-b741-fa45a61..." in US East (N. Virgi...



6. You can also check **CloudWatch Alarms** and get more details on CloudWatch console:



7. Alternatively, you can have AWS Lambda as the subscriber endpoint, so as soon as the threshold is breached, SNS will call the lambda function, which in turn will disable the workgroup and preventing from executing further queries within that workgroup. Feel free to explore multiple subscriber endpoints.

Cost Allocation Tags

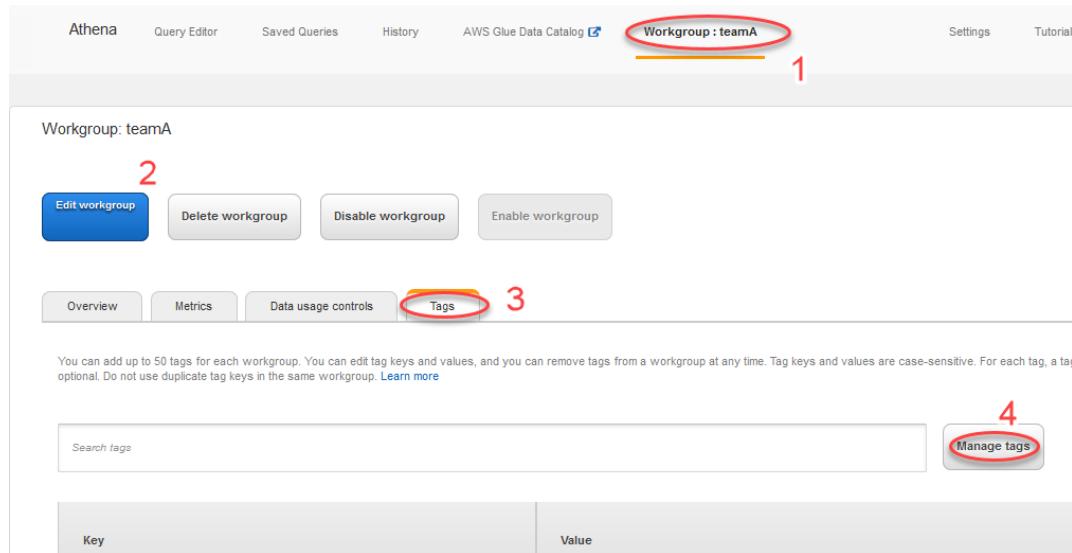
When you created two workgroups: **workgroupA** and **workgroupB**, you also created **name as tags**. These tags can be utilized in Billing and Cost Management console to determine the usage per workgroup.

For example, you can create a set of tags for workgroups in your account that helps you track workgroup owners, or identify workgroups by their purpose. You can **view tags for a workgroup** in “View Details” page for the workgroup under consideration.

You can add tags later after you have created workgroup. To create tags:

1. Open the Athena console at <https://console.aws.amazon.com/athena/>, choose the **Workgroups** tab, and select the workgroup.
2. Choose **View details** or **Edit workgroup**.
3. Choose the **Tags** tab.

4. On the **Tags** tab, choose **Manage tags**, and then specify the key and value for each tag.
5. When you are done, choose **Save**.



For more details on best practices:

<https://docs.aws.amazon.com/athena/latest/ug/tags-console>