**Creating an NBA Data Lake for Analytics Using AWS Services**

In this project, I created an NBA data lake to store and analyze NBA-related data using AWS services. The main tools I used are Amazon S3, AWS Glue, and Amazon Athena. Let me walk you through the process and the key components of this project.

**The Goal:**

The goal was to set up a data lake for NBA data and create an analytics pipeline that would allow us to easily store, process, and query NBA data.

**How It Works:**

1. **Amazon S3 (Storage):**
   * First, I created an **S3 bucket**. This bucket acts as the storage system for both raw and processed data.
   * Then, I uploaded sample NBA data in **JSON format** to this S3 bucket. This data contains information like player stats, teams, and more.
2. **AWS Glue (ETL and Data Catalog):**
   * Next, I set up **AWS Glue** to create a **database** and an **external table** to manage and query the data.
   * AWS Glue also helps in transforming raw data into a format that’s easy to query.
3. **Amazon Athena (Querying Data):**
   * With **Amazon Athena**, I enabled querying of the data stored in S3 without the need to set up a separate database server.
   * Athena allows me to run SQL-like queries directly on the data stored in S3, and I can get analytics results right away.

**Step-by-Step Breakdown:**

**Step 1: Accessing Your AWS Account:**

* First, you sign into your AWS Console. You can either use AWS CloudShell for quick access or authenticate via AWS CLI from your terminal.

**Step 2: Setting Up the Script:**

* I wrote a Python script (setup\_nba\_data\_lake.py) that automates the creation of all the necessary AWS resources.
  + The script creates an S3 bucket.
  + It uploads sample NBA data (JSON format) into S3.
  + It sets up an AWS Glue database and an external table for querying.
  + It configures Amazon Athena for querying the data.

**Step 3: API Access (For Real Data):**

* To get real-time data, I integrated an API from **Sportsdata.io**. This API gives access to NBA data like player stats, standings, and scores.
* I registered for the API, obtained an API key, and used it in the Python script to pull data.

**Step 4: Running the Script:**

* Once everything was set up, I ran the Python script, and it automated the whole process:
  + Created the S3 bucket.
  + Uploaded the NBA data.
  + Configured the AWS Glue database.
  + Set up Athena to query the data.

**Step 5: Verifying the Resources:**

* After running the script, I verified the resources in the AWS Console.
  + In **S3**, I checked if the data was uploaded correctly.
  + In **Athena**, I ran a simple SQL query to pull data on NBA players, like fetching all point guards (PG), and confirmed that the data was queryable.

**Cleanup:**

* To ensure everything could be deleted properly after the project, I created a cleanup script (delete\_resources.py). This script:
  + Deletes the S3 bucket and all of its contents.
  + Removes the AWS Glue database and tables.
  + Clears any Athena configurations.

**Key Learnings:**

* **AWS Permissions and Security:** I learned how to secure AWS services using the principle of least privilege for IAM roles and policies.
* **Automating AWS Resources with Python:** I automated the process of creating cloud resources, like S3, Glue, and Athena, using Python scripts.
* **Integrating External APIs:** I also learned how to integrate external APIs into a cloud workflow to bring real-time data into the AWS environment.
* **Data Processing and Querying:** Finally, I got hands-on experience processing raw data in S3 and querying it using Amazon Athena.

**Why It’s Useful:**

* This setup allows us to store vast amounts of NBA data, run analysis directly on that data using SQL queries, and even bring in real-time data using an API. This type of data lake architecture is ideal for handling large datasets and running analytics without worrying about complex infrastructure.

So in a nutshell, this project helped me understand how to build a data lake on AWS, integrate APIs for real-time data, and use cloud services like S3, Glue, and Athena to store, process, and query data efficiently. It's a great way to store large datasets and make them easily accessible for analysis.