**Connecting Power BI to Amazon Athena**

**Things to consider:**

When granting permissions in AWS, follow the principle of least privilege. Only grant necessary permissions to reduce the attack surface and minimize the impact of security breaches. For powerful permissions like Amazon Athena and S3 Full Access, consider key factors to ensure the least privilege.

1. **Specific Access Needs**: To grant access to appropriate resources, identify user needs and actions required. For Amazon Athena Full Access, decide on admin or limited permissions. For Amazon S3 Full Access, specify the buckets and actions needed.
2. **Separation of Concerns:** Grant separate permissions for different tasks to reduce the risk of unintended actions or data exposure.
3. **Resource Isolation:** Isolate resources and restrict access only to what's needed. For Amazon Athena, create dedicated databases and tables for different users or apps with access controls. Amazon S3 uses bucket policies and ACLs to limit access to specific buckets and objects.
4. **Least Privilege IAM Policies**: Create IAM policies with minimal permissions required for intended tasks. Avoid granting blanket access to all resources. Instead, specify exact actions and resources needed. Regularly review and update policies as access requirements change.
5. **Monitoring and Logging:** Use logging and monitoring to detect unauthorized activity. AWS provides services like CloudTrail and CloudWatch to track user actions and alert you of potential security threats.
6. **Regular Reviews and Audits**: Conduct regular reviews and audits of permissions to ensure they align with business requirements and follow the principle of least privilege. This includes reviewing IAM policies, S3 bucket policies, and other access controls to identify and remediate any overly permissive permissions.

Regarding the creation of an Athena Output Bucket and granting access to the user, follow these steps:

1. **Create an Athena Output Bucket:** Create an S3 bucket specifically for storing query results generated by Athena. This bucket should have appropriate encryption, versioning, and access control settings based on your organization's security policies.
2. **Grant Access to the User:** Once the bucket is created, configure the necessary permissions to allow the user to write query results to the bucket. This typically involves updating the bucket policy or IAM policies associated with the user to grant the s3:PutObject permission for the Athena Output Bucket.
3. **Test Access:** Before deploying any production workloads, thoroughly test the user's access to ensure they can successfully write query results to the Athena Output Bucket without encountering permission errors.
4. **Regularly Review Access:** As with any permissions configuration, regularly review and audit the user's access to the Athena Output Bucket to ensure it remains aligned with business requirements and follows the principle of least privilege. Remove any unnecessary permissions or access that is no longer required.

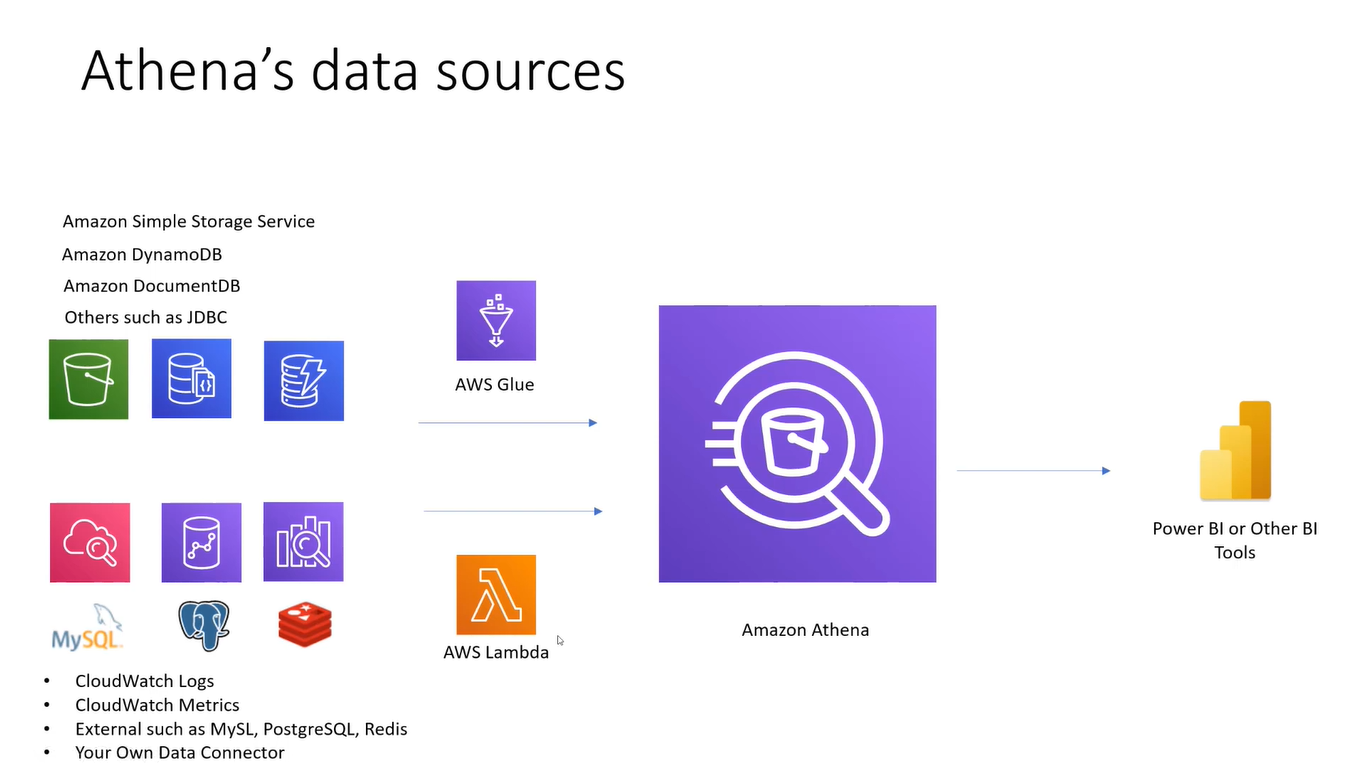
**Athenas data sources**

Amazon Simple Storage Service

Amazon DynamoDB

Amazon Documentation DB

Others such as JDBC



**There are three components.**

Data Source

Catalog

Athena Query Engine

A screenshot of a computer

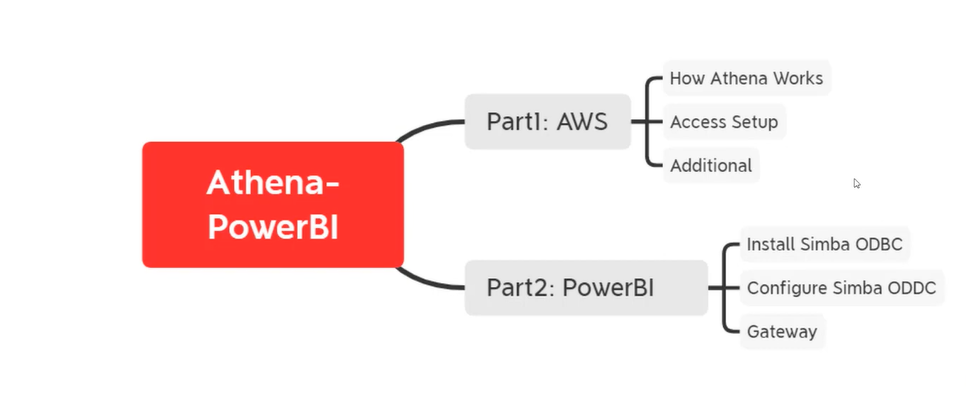
Description automatically generated

**For Advance Authentication: Instance Profile**

To establish a more secure authentication method for accessing Power BI, it's recommended to utilize an Amazon EC2 instance as the workspace or Power BI Gateway server. This ensures that access is controlled within the AWS ecosystem. The Power BI Admin responsible for setting up the on-premise gateway can create an EC2 Role with the necessary permissions, such as accessing S3 buckets or DynamoDB tables.

These permissions are then attached to the EC2 Role without the need for access keys, enhancing security by eliminating the risk associated with key management. Once the EC2 Role is created, it can be easily attached to the EC2 instance running the Power BI Gateway server, allowing seamless integration with AWS services while maintaining a robust security posture. Regular monitoring and access reviews should be conducted to ensure the continued effectiveness of this security setup.

**Let’s connect Athena to Power BI**

****

**Part 1: AWS**

**Things to consider.**

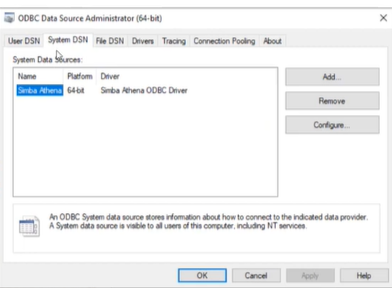
Also confirm if you have received the region, output bucket location, database, Athena workspace

Given the serverless nature of Athena, it's important to note that performance and download speed optimizations are not configurable directly. However, optimizing underlying data storage in S3, partitioning data effectively, and optimizing queries can indirectly improve query performance.

Lastly, since custom queries are not permitted, collaboration with the Power BI Admin or someone with access to the Athena console is necessary to create the required views or manage Athena configurations effectively. Clear communication and coordination with the admin ensure smooth operations within the established constraints.

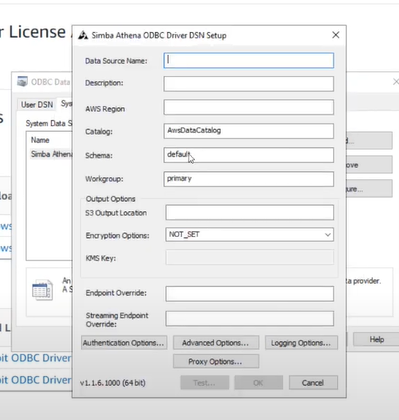
To download the ODBC here is the link [click here](https://docs.aws.amazon.com/athena/latest/ug/connect-with-odbc.html)

1. Upon download completion, launch the ODBC driver and create a new data source.
2. Navigate to System DNS and select the "Add" button.



1. Navigate to System DNS and select the "Add" button. In the dialog box, locate the downloaded data source within the list and select it, then proceed by clicking on "Finish".A screenshot of a computer

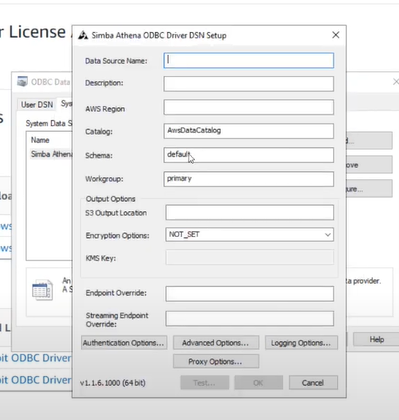
   Description automatically generated
2. Once you have clicked on the "Finishing" button, a window will appear on your screen displaying the following information.

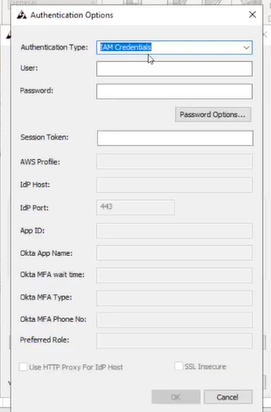


1. Please ensure that you provide the name and description of the data source.
2. However, the region, output, and S3 output location should be provided by your AWS administrator.
3. The schema and catalog should be kept as default, although the schema is the database name that you need to input into the information.

A screenshot of a computer

Description automatically generated

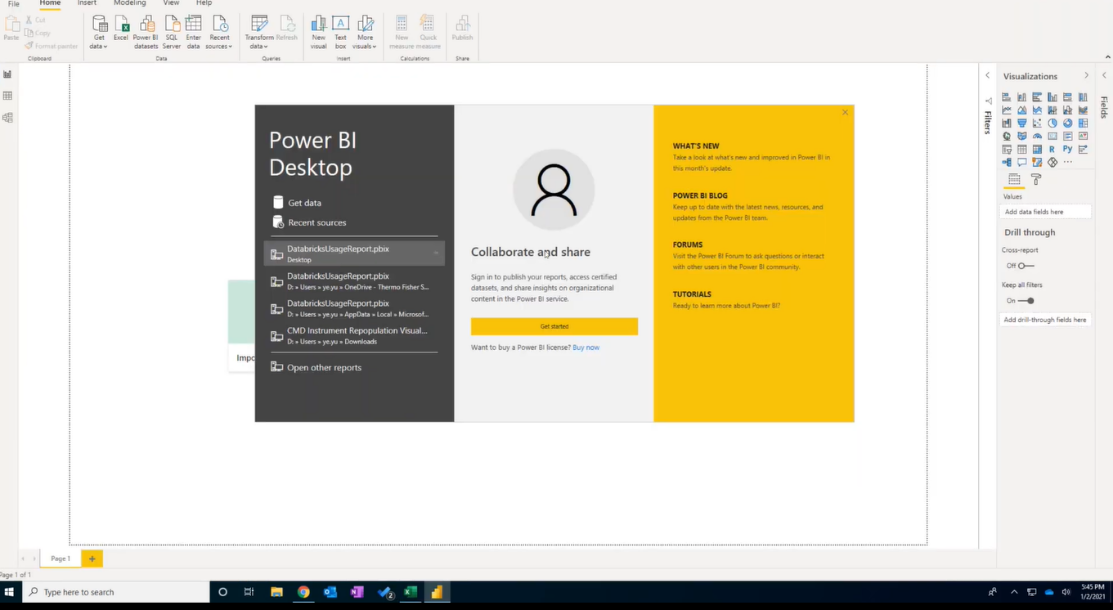
1. Please click on "authentication options" and then proceed to obtain an access key.
2. 
3. In the User and Password section, please type the access key provided by your administrator.



1. If you have connected your database correctly, you will receive a success message prompt. Just click "OK".

**Part 2: Power BI**

1. First of all open the Power bi desktop and click on Get data source



1. To connect to ODBC, please search for it and click on the 'connect' option.A screenshot of a computer

   Description automatically generated
2. After clicking on "connect", you will have the option to select your database. Once you have selected the database, click "OK".
3. To access the system, you will be required to provide your username and password. Please make sure you have both pieces of information ready before proceeding.

A screenshot of a computer

Description automatically generated

1. From the Navigator pane, you can access all the data such as tables, and select specific data to upload to Power BI.

**What are the best practices for securing a database application with ODBC or JDBC drivers?**

## Use encryption

One of the most important steps to secure your database application is to use encryption for both the data in transit and the data at rest. Encryption protects your data from unauthorized access, modification, or theft. You should use strong encryption algorithms and protocols, such as AES, SSL, or TLS, to encrypt your data before sending or storing it. You should also use secure keys and certificates to authenticate your encryption and store them in a safe location. You should avoid using plain text or weak encryption methods, such as MD5 or SHA-1, as they are vulnerable to attacks.

## Limit privileges

Another best practice for securing your database application is to limit the privileges of your ODBC or JDBC drivers, as well as the users and applications that access them. You should follow the principle of least privilege, which means granting only the minimum permissions and roles necessary for your database operations. You should also use different drivers and accounts for different purposes, such as development, testing, and production, and segregate them by network zones or firewalls. You should avoid using default or generic drivers and accounts, as they may have excessive or unknown privileges that could compromise your data.

## Validate input

A third best practice for securing your database application is to validate the input that you pass to your ODBC or JDBC drivers. Input validation is the process of checking the data that you receive from users or other sources for errors, malicious code, or unexpected values. You should use proper data types, formats, and lengths for your input, and sanitize or escape any special characters that could cause SQL injection or other attacks. You should also use prepared statements or parameterized queries, which separate the data from the SQL commands, and prevent the execution of arbitrary code.

## Monitor activity

A fourth best practice for securing your database application is to monitor the activity of your ODBC or JDBC drivers, as well as the data sources that they connect to. Monitoring activity is the process of collecting, analyzing, and reporting information about the performance, usage, and behavior of your database components. You should use tools and techniques, such as logging, auditing, alerting, or tracing, to track and record the events and transactions that occur in your database environment. You should also review and update your security policies and procedures regularly, and respond to any anomalies or incidents promptly.

## Secure configuration

A fifth best practice for securing your database application is to secure the configuration of your ODBC or JDBC drivers, as well as the data sources that they connect to. Secure configuration is the process of setting up and maintaining the optimal settings and options for your database components. You should use best practices and standards, such as OWASP, NIST, or CIS, to guide your configuration decisions. You should also disable or remove any unnecessary or unused features, services, or ports that could expose your data to risks. You should also update and patch your drivers and data sources frequently and test them for vulnerabilities.

## Educate users.

A sixth and final best practice for securing your database application is to educate the users and developers who interact with your ODBC or JDBC drivers. Education is the process of providing training, awareness, and guidance on the security principles and practices that apply to your database environment. You should inform your users and developers about the risks and responsibilities that come with using ODBC or JDBC drivers, and the best ways to protect their data and credentials. You should also encourage them to report any issues or concerns that they encounter and provide feedback and support to improve your security posture.