# Export RDS and EC2 SQL Server Audit Logs to Centralized Data Lake for Compliance, Audit and Analysis

*Swarndeep Singh, DB Migration Consultant*

Most of the commercial database engines provide auditing solutions for compliance and regulations. When it comes to SQL Server it’s no exception, SQL Server provides auditing solutions to capture database activities. On AWS SQL Server on EC2 and RDS both support these auditing features. Out of the box, SQL Server keep the auditing data to local resources like audit files, event logs, etc. Keeping logs on local systems are fine as long as there are few number of SQL Server, it solves the purpose to keep the auditing data locally. However, in case large environments with more than dozens of SQL Servers it would make more sense to centralize the auditing data at once place for all of the SQL Servers.

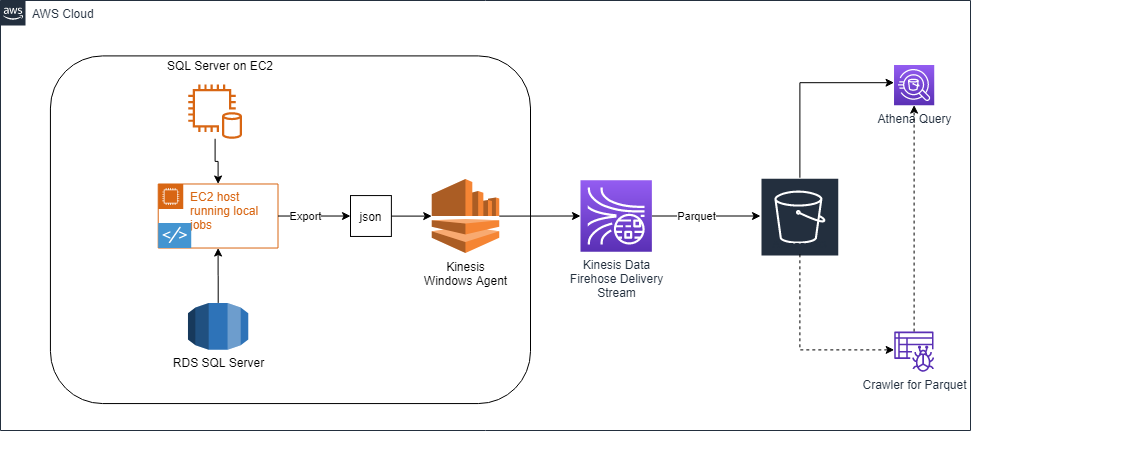
This blog post demonstrates the solution to centralize the auditing data from RDS and EC2 SQL Servers and keep it at single destination on S3 in Parquet format on AWS and makes it searchable using Athena queries.

# SQL Server Auditing

SQL Server supports Auditing at instance level and database level that involves logging events based on your requirement. Setting up a SQL Server Auditing on EC2 requires you to create server audit and then server level audit specification and database level audit specification. In audit specification you define events to log in event file. More information about setting up auditing [here](https://docs.microsoft.com/en-us/sql/relational-databases/security/auditing/sql-server-audit-database-engine?view=sql-server-ver15) [https://docs.microsoft.com/en-us/sql/relational-databases/security/auditing/sql-server-audit-database-engine?view=sql-server-ver15]

In Amazon RDS you create the audit and audit specification in the same way that you create them on –premises or EC2. More information about setting up auditing on RDS [here](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.SQLServer.Options.Audit.html#Appendix.SQLServer.Options.Audit.AuditRecords) [https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.SQLServer.Options.Audit.html#Appendix.SQLServer.Options.Audit.AuditRecords]

# Solution Overview



In this solution, User creates server and database audit specifications. SQL Server creates audit files in SQL Server native sqlaudit format. Using PowerShell scripts, the sqlaudit files output is converted to JSON format and saves it to a predefined folder.

A Kinesis Windows agent is configured to read the JSON data as exported above and then transfers to Kinesis Data Firehose Deliver stream. The Kinesis Data Firehose Delivery Stream is configured to convert the incoming stream to parquet format and save it to S3 bucket. Finally, Athena service lets you create a data source using crawler and you can then query the data directly from S3 bucket.

In this demonstration, there will be two sources for auditing, an EC2 SQL Server and an RDS instance.

# Setting up the auditing solution

## Prerequisites:

1. AWS Account
2. Amazon EC2 key pair
3. Amazon EC2 Instance with SQL Express Edition (In this demonstration, there will be one EC2 instance for dual purpose, running a SQL Server Instance and Executing PowerShell scripts).
4. Amazon S3 Bucket
5. Amazon Kinesis Data Firehose Delivery Stream.
6. Amazon RDS for SQL Server
7. Amazon Kinesis Agent for Microsoft Windows.
8. Amazon Athena
9. AWS Glue Database and Crawler.
10. Github repository

## Here are the steps to setup the solution:

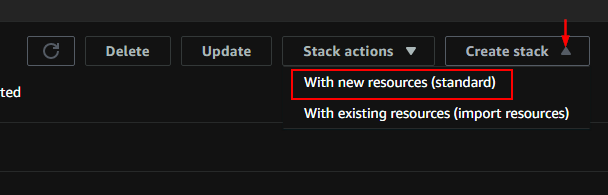
1. Download the GitHub solution.
2. Deploy the Stack from the Cloud Formation Scripts from github. Cloud formation would create the following components.
   1. S3 Buckets to store audit logs and WorkGroup cache.
   2. Kinesis Data Firehose Delivery Stream
   3. EC2 instance (This can be a dual purpose EC2 instance running SQL Server instance or can be a separate instance. In this example it is a dual purpose EC2 instance).
   4. RDS instance for SQL Server.
   5. AWS Glue Database.
   6. Amazon Athena WorkGroup
   7. IAM Roles
3. Install Kinesis Windows Agent on EC2 instance
4. Configure Kinesis Agent for Windows
5. Setup PowerShell script on EC2 instance to convert sqlaudit to JSON format.
6. Configure environment
7. Setup SQL Audits
8. Setup Athena to query audit data from S3 Data Lake (parquet).

### Download solution from Github

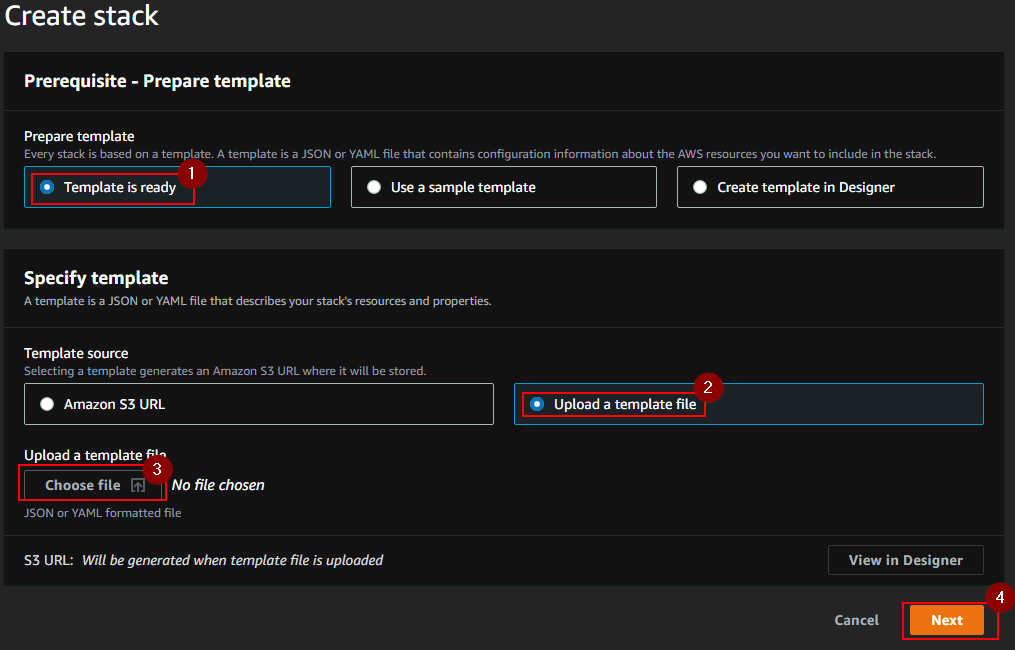
Download github solution from <https://github.com/swarndeepsingh/Auditing-Blog.git>.

### Deploy the Stack using the Cloud Formation

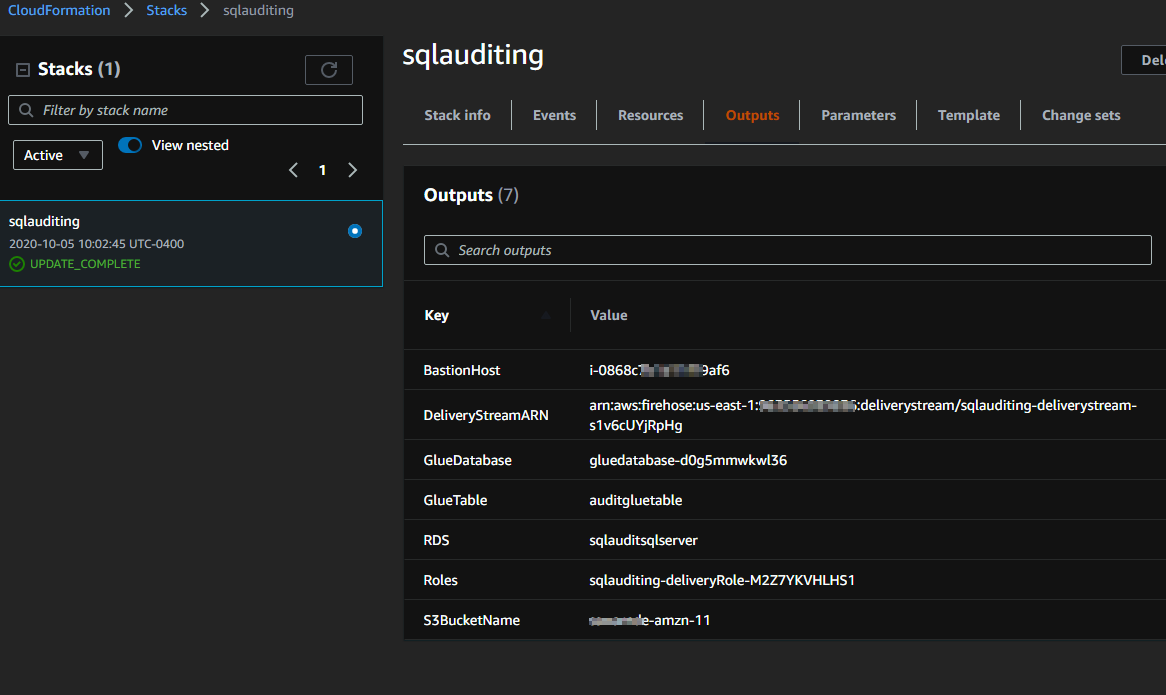
1. From the github solution that was mentioned above, navigate to \aws-sql-migration-automation\auditing\scripts\cf\audit-setup.yml cloud formation file and refer the audit-setup.yml yml file.
2. Open CloudFormation service from AWS Console.
3. From Create Stack drop down, select with new resources (standard)



1. From Create stack page, select “Upload template file” and “Choose file” audit-setup.yml.



1. Click Next and follow the wizard to complete the stack creation.
2. Once stack completes, click on Outputs tab of the stack created previously.



### Install Kinesis Windows Agent on EC2 instance

1. Logon to EC2 instance that was created as part of the CF Stack.
2. Please follow the steps as mentioned in [this user guide](https://docs.aws.amazon.com/kinesis-agent-windows/latest/userguide/getting-started.html#getting-started-installation) to install the Kinesis Agent for Windows.
3. You would have to allow Security Group rule to allow RDP to the EC2 instance. Please follow this [link](https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/authorizing-access-to-an-instance.html) to know more about adding rules for RDP.

### Configure Kinesis Agent for Windows

1. [Configuring Kinesis Agent](https://docs.aws.amazon.com/kinesis-agent-windows/latest/userguide/directory-source-to-s3-tutorial.html) requires modifying the configuration file and provide information for Source, Sink and Pipe declarations.
2. From EC2 host, open file explorer and open file C:\Program Files\Amazon\AWSKinesisTap\appsettings.json in a text editor. Replace the contents as shown below, and modify the parameters as with appropriate values as shown in example below:
   1. Directory: Create a folder on EC2 and update in the parameter.
   2. StreamName: From CloudFormation output tab, copy the DeliveryStream Name and copy in this parameter.
   3. Region: Change the region name based on AWS configuration.

{

"Sources": [

{

"Id": "JsonLogSource",

"SourceType": "DirectorySource",

"RecordParser": "SingleLineJson",

"Directory": "C:\\LogSource\\copy\\",

"FileNameFilter": "\*.log",

"InitialPosition": 0

}

],

"Sinks": [

{

"Id": "FirehoseLogStream",

"SinkType": "KinesisFirehose",

"StreamName": " sqlauditing-deliverystream-s1v6cUYjRpHg",

"Region": "us-east-1",

"Format": "json",

"ObjectDecoration": "ComputerName={ComputerName};DT={timestamp:yyyy-MM-dd HH:mm:ss}"

}

],

"Pipes": [

{

"Id": "JsonLogSourceToFirehoseLogStream",

"SourceRef": "JsonLogSource",

"SinkRef": "FirehoseLogStream"

}

]

}

1. On EC2, using elevated PowerShell session run the following command to start the Kinesis Agent Windows Service
   1. Start-Service –ServiceName AWSKinesisTap

### Setup PowerShell script on EC2 instance to convert sqlaudit to JSON format

1. From the content downloaded from Github, refer to \aws-sql-migration-automation\auditing\scripts\powershell\audit\_json.ps1 and schedule it from Windows Task Scheduler to run based on your preference. You need to pass the configuration file (config.json) full path as an argument to this the script.

Note: You need to schedule this for each server separately by passing appropriate config file. More details on config file is provided in next step. In this case since there are two servers and two config files, thus you will have to setup two jobs. You may schedule the job to run based on preferences, like hourly or every few hours etc.

Example:

C:\aws\aws-sql-migration-automation\auditing\scripts\powershell\audit\_to\_json.ps1 C:\script\config-ec2.json

C:\aws\aws-sql-migration-automation\auditing\scripts\powershell\audit\_to\_json.ps1 C:\script\config-rds.json

### Configure Environment

1. You need to provide parameters for config file used by this process. Each file is mapped with one server, in order to have multiple servers being monitored, you need to add multiple config files. (located at \aws-sql-migration-automation\auditing\scripts\config). In this demonstrator there are two servers being monitored (SQL Server on RDS and SQL Server on EC2). Please see the parameters as used the configuration file:

**Type**: Takes EC2 or RDS as parameter to specify the target SQL Server whether It’s EC2 instance or RDS Instance.

**Sqlserver**: Instance name of the remote SQL Server that needs to be audited.

**Dbname**: This is name of the database that will be created on each SQL Server to track the auditing data.

**Auditdata**: This is the location on remote SQL Server host where the audit files will be stored. On EC2, you need to create this path yourself, however, on RDS, this parameter will be ignored.

**Auditout**: This is the location where JSON converted audit data will be stored. This path must be the same path for appsettings.json LogSource for Kinesis Data Input, in this case C:\\LogSource\\copy\\".

**Sqlscriptpath**: This path contains the sql server objects which are required to created first time.

**Auditname**: This is the master audit name.

**Dbauditname**: This is the Database Audit Specification name.

**Serverpsecaudit**: This is the Server Audit Specification name.

Examples:

File: Config-ec2.json:

"config":[

{

"sqlserver":"EC2SQLServer "

},

{

"type":"ec2"

},

{

"dbname":"awsaudit"

},

{

"auditdata":"c:\\aws\\auditdata"

},

{

"auditout":"C:\\LogSource\\copy"

},

{

"sqlscriptpath":"C:\\aws\\aws-sql-migration-automation\\auditing\\scripts\\sql"

},

{

"auditname":"masteraudit"

},

{

"dbauditname":"databaseaudit"

},

{

"serverspecaudit":"serveraudit"

}

]

}

File: Config-rds:

{

"config":[

{

"sqlserver":"sqlauditsqlserver.cfrxbkwbf0bo.us-east-1.rds.amazonaws.com"

},

{

"type":"RDS"

},

{

"dbname":"awsaudit"

},

{

"auditdata":"c:\\aws\\auditdata-rds"

},

{

"auditout":"C:\\LogSource\\copy"

},

{

"sqlscriptpath":"C:\\aws\\aws-sql-migration-automation\\auditing\\scripts\\sql"

},

{

"auditname":"materaudit"

},

{

"dbauditname":"databaseaudit"

},

{

"serverspecaudit":"serveraudit"

}

]

}

### Setup SQL Audits

1. Execute PowerShell script “setupdbconfiguration.ps1” from \aws-sql-migration-automation\auditing\scripts\powershell to setup the SQL procedures, tables and audit specification. You need to pass the configuration file (config.json) full path as an argument to this script. This needs to be executed for each server that needs to be audited by passing the config file path.

Example: C:\aws\aws-sql-migration-automation\auditing\scripts\powershell\setupdbconfiguration.ps1 C:\script\config.json

Above scripts will create the database, tables and DB and server audits. You may modify the DB and Server Audit specification based on your requirements, either from using T-SQL scripts or directly using SSMS after executing of the PowerShell file.

### Setup Athena

1. The Cloud Formation creates Glue Workgroup that includes querying the data from S3 parquet format.
2. Once setup, you can run query by selecting the custom workgroup as shown below and extract the audit data based on custom filters as needed.

