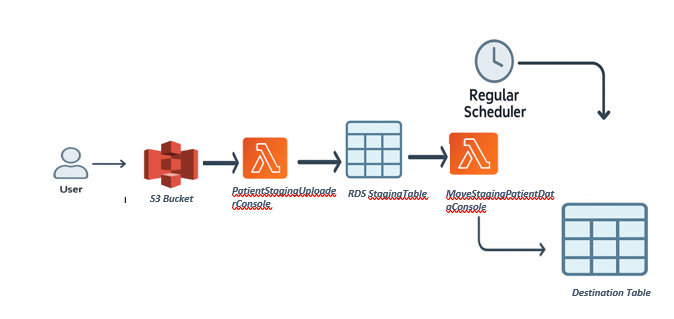
**Content:**

1. System Flow Diagram
2. AWS CLI Installation
3. Aws Configure
4. Create S3 Bucket
5. Create RDS Database
6. Add Inbound Rule to the Security Group
7. Enable Internet Access to the Lambda Function
8. What to check Lambda-RDS Connectivity
9. Significance of NAT – Gateway
10. Subnet Types
11. Private Route Tables
12. Create NAT – Gateway
13. Attach Lambda to private subnets
14. Create staging table local SSMS
15. Create Lambda Function in .Net Framework
16. Deploy Lambda Function
17. Monitor Lambda Execution
18. How to Debug Lambda Function on Local Machine
19. Final Step Verification



**2.To work in AWC CLI:**

Install AWSCLIV2 from [Installing or updating to the latest version of the AWS CLI - AWS Command Line Interface](https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html).

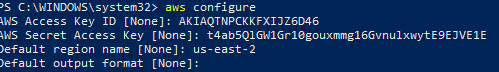
* Install **AWS CLI v2** from the official AWS documentation.
* Ensure the installation path: C:\Program Files\Amazon\AWSCLIV2 (aws.exe)
* Verify the installed version.



**3. AWS Configure :**

Create Access Key and Secret Access Key.

1. **Log in to the AWS Console** 🔗 <https://console.aws.amazon.com/>
2. In the top-right corner, click your **username** → choose **“Security credentials”**
3. Scroll to the **Access keys** section
4. Click **“Create access key”**
5. Choose **"Command Line Interface (CLI)"** as the use case, and click **Next**
6. Give it a name/description and click **Create access key.**



**Setting Up AWS Resources**

**4. Create S3 Bucket:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Public Access** | **Region** | **ACLs disabled** | **Bucket Versioning** | **Bucket Key**: |
| patientinfo-json-staging | Block all public access. | Us-east-2 | **True** | Disabled | Enabled |

Enabled default encryption

5. **Create RDS Database:**

* Choose database engine(Sql Server)
* Leave as is about the remaining option.
* Select **Free Tier** template
* Database Instance Name
* Define credentials (**Username & Password** or enable Windows Authentication).

Choose VPC Security Group: either existing or new: (Note: same VPC must be choose for the lambda function since it uses same data base instance. So, both should run under same VPC and security group)

It will create below configuration for DB instance:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Make sure **"Public access"** is **set to "Yes":**

If the RDS instance is **private**, your **local machine** cannot access it directly unless there is a **VPN** or **Direct Connect** in place. If your RDS instance is **public**, you can connect to it using its **public endpoint**.

**6.** **Add Inbound Rule to the Security Groupe:**

**Steps:**

1. Go to **EC2 > Security Groups**
2. Find the security group attached to your RDS instance
3. Click **Inbound rules > Edit inbound rules**
4. Add this rule:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type** | **Protocol** | **Port Range** | **Source** | **Source** |
| MSSQL/Aurora | TCP | 1433 | Custom | 7 |
| MSSQL | TCP | 1433 | Custom | SecurityGroup ID |

**7.Use of NAT Gateway:**

NAT Gateway is required when lambda function needs to connect with internet resources . If Lambda only needs to access internal VPC resources like RDS instance , then no NAT Gateway is required.

**8. Subnet Types:**

Need 2 types of subnets:

**Public Subnet**

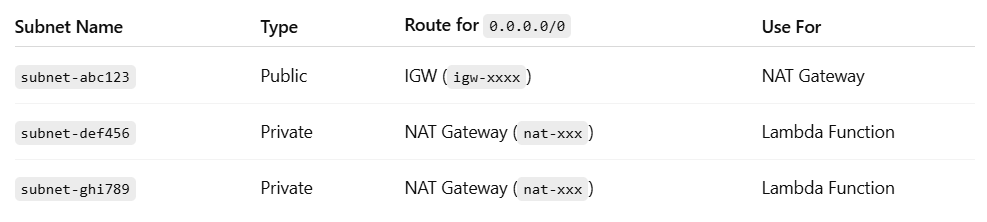
* Has route to Internet Gateway (igw-xxxx)
* Hosts the NAT Gateway
* Route Table: Destination: 0.0.0.0/0 → Target: igw-xxxxxxxx (Internet Gateway)
* **Internet Access**: Instances can directly access the internet.

**Private Subnet**

 Has route to **NAT Gateway (nat-xxxx)** Lambda function should use this  
 Route Table: Destination: 0.0.0.0/0 → Target: nat-xxxxxxxx (NAT Gateway)  
 **Internet Access**: Instances cannot directly access the internet.

**Subnets classification for current scenario:**

By default there are 3 subnets will be created for a VPC. Pick one to remain public . This subnet will host the NAT Gateway.



**9.Create Private** **Route Table:**

* Go to **VPC > Route Tables > Create Route Table**
* Name it private-route-table
* Associate it with the **other two private subnets** (e.g. subnet-00ffcb8e38c12d55a, subnet-07a51761582d88a75)
* Do **not** add a route to the Internet Gateway

**10. Steps to Add a NAT :**

**Create a NAT Gateway**:

* Go to the **VPC Console** in the AWS Management Console.
* Under **NAT Gateways**, click **Create NAT Gateway**.
* Select a **public subnet** to associate the NAT Gateway
* Ensure that **VPC route table** directs outbound traffic from the private subnets to the **NAT Gateway**.

**Add NAT route to private route table**

Edit the **private-route-table**

Add route:

**Destination: 0.0.0.0/0**

**Target: nat-xxxxxxxx (your new NAT Gateway)**

#### ****11.Attach Lambda to private subnets****

Update your Lambda configuration:

* VPC: same VPC
* Subnets: the two that now use the private route table
* Security group: ensure it allows outbound to port 443 (for HTTPS)

**Update Route Tables**:

* Once the NAT Gateway created, update the **private subnet's route table** to route outbound traffic (0.0.0.0/0) to the NAT Gateway. This will allow the Lambda function in the private subnet to access the internet.

**Lambda Internet Access**:

Once the NAT Gateway is in place, Lambda functions in the private subnet will be able to access external services via the internet (To connect Sql Server from local machine to RDS DB Instance ).

**12.Create staging table locally:**

connect to the RDS database instance remotely and execute the script there

There are 2 options from local machine using a DB client like:

* **SQL Server**: SQL Server Management Studio (SSMS) or Azure Data Studio

Ex:

|  |  |  |  |
| --- | --- | --- | --- |
| **Server name**: | **Authentication** | **Login** | **Password** |
| stagingdb.c3yeo4ksyngm.us-east-2.rds.amazonaws.com,1433 *(Note:1433 is a port for sql server and Endpoint is available on the RDS db instance page)* | SQL Server Authentication | RDS master username | RDS master password |

**To test the connection succeed**

**Option 1:**

On RDS dashboard , the below points have to be confirmed

 The DB instance status is **"Available"**

 The endpoint matches

**Option 2:**

Powershell:

Test-NetConnection stagingdb.c3yeo4ksyngm.us-east-2.rds.amazonaws.com -Port 1433

**Create table on local SSMS:**

SQL script for staging table in RDS DB once connected from SSMS to the AWS

CREATE TABLE PatientStaging ( Id INT PRIMARY KEY,

FullName NVARCHAR(100), DateOfBirth DATE,

Email NVARCHAR(100), Phone NVARCHAR(50),

HcpSpecialty NVARCHAR(100), Indication NVARCHAR(100),

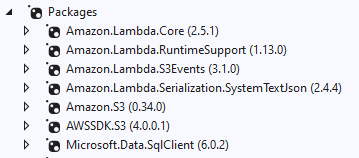
InsuranceType NVARCHAR(100), ConsentToEmail BIT,

ConsentToText BIT, PinHash NVARCHAR(256), CopayCardNumber NVARCHAR(50));

**13. Create Lambda Function:**

Created Lambda function in .Net framework(version 8.0 -as some aws lambda packages are in lower version) locally. It will trigger when staging json file uploaded in to S3 bucket.

Install necessary packages.



Create lambda function in separate class file for Lamda function and invoke it using AWS Lambda **RuntimeSupport** package to define and invoke it.

**14.Deployment:**

Deploy Lambda function on Powershell:

dotnet lambda deploy-function ProjectName --region us-east-2 --runtime --function-handler Assembly::Namespace.Function::FunctionHandler.

Ex: dotnet lambda deploy-function MoveStagingPatientDataConsole --region us-east-2 --runtime --function-handler MoveStagingPatientDataConsole ::MoveStagingPatientDataConsole.Function::FunctionHandler.

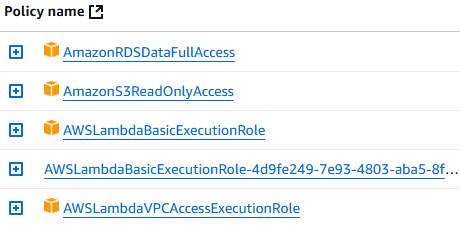
Once it deployed successfully It will be populated on AWS/Lambda console

Lambda Configuration on AWS:

On Configuration Tab=>

* + General Configuration: Set the timeout limit. Default is 30 sec and It will allow upto 15 minutes
  + Permissions: map the IAM role to access s3, rds .

It should have default below policies attached for S3, RDS and CloudWatch Logs.



VPC: Same VPC using on RDC DB

RDS DB: RDS DB should be map here

Setup S3 trigger for Lambda Function “PatientStagingUploaderConsole”:

 Go to **AWS Lambda Console** →  **Lambda Function** → **Designer** section.  
 Click on **+ Add Trigger**.  
 Choose **S3** as the event source.  
 Select the S3 bucket and specify the event type (e.g., "ObjectCreated").  
 Click on **Add**.

Now, the Lambda function “PatientStagingUploaderConsole” will be triggered every time a new file is uploaded to the selected S3 bucket.

**15.Enable inter access for Lambda once deployed**

* Attach a **NAT Gateway** to the **public subnet** of your VPC. It is to connect SQL Server from local machine when Lambda function insert data from S3 into staging table.
* Ensure the **private subnet** (where Lambda runs) has a **route to the NAT Gateway**.

**16.What to Check Next for Lambda–RDS Connectivity**

* Lambda function is attached to this same VPC and subnet  
   Select at least 2 **private subnets** that have access to the RDS
* Lambda security group allows outbound access to port 1433 (for SQL Server)
* RDS security group group allows Sql Server from local machine to connect with RDS DB instance through inbound rule of security group by allocate public IP of local machine to the port 1433 and TCP protocol.
* Database endpoint and credentials are correct

**17.Monitor Lambda Execution:**

**CloudWatch Logs**: Lambda automatically sends logs to CloudWatch, so check logs for debugging purposes or performance monitoring.

* Log Groups available **on**  **AWS CloudWatch Console** → **Logs** → **Log Groups** → Find the log group for PatientStagingUploaderConsole.

**18.How to Debug Lambda Function on Local Machine:**

Below packages have to be installed locally on C:\Users\.dotnet\tools and project directories using powershell.

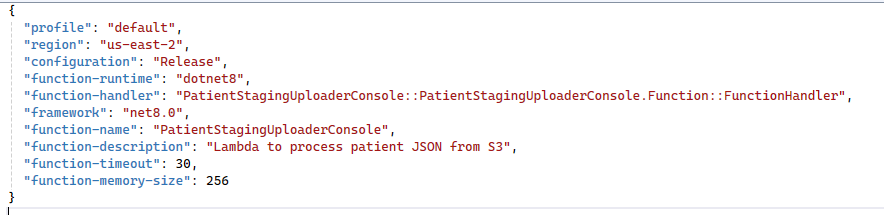
Before that Add this path to the Environment Variable (Search Environment Variable => Path=>Edit=>New=>Add C:\Users\.dotnet\tools). So that the .Exe file will be installed to this path.

**dotnet tool install -g Amazon.Lambda.Tools**

**project directory : dotnet-lambda-test-tool-8.0**

It will create localhost server <http://localhost:port>

Below is the configuration file . it should be deployed on aws and selected when test the lambda function on local server.



### **19.Final Verification Steps for Inbound and Outbound(Only Lambda function connect local resource like RDS instance):**

1. **Subnet Selection (within the VPC):**
   * Ensure Lambda Function is attached to **private subnets** that have access to the **RDS instance**.
   * The subnets **must have a route to a NAT Gateway**
2. **Security Groups:**
   * The Lambda's **security group** must allow **outbound traffic HTTPS** (port 1433).

* The **RDS security group** must allow **inbound traffic on port 1433** from the **Lambda's security group** (or its IP range).
* **NAT Gateway:** Located in a **public subnet** (with IGW route) and Associated **Elastic IP**.

1. **RDS Endpoint and Port:**
   * Double-check that Lambda code is using the **correct RDS endpoint** (not the public one if the RDS is private-only).
   * Ensure port **1433** is included in the connection string.

4. **Lambda Logs**

* Watch **CloudWatch Logs/ Log groups** for a specific Lambda function execution and errors.(<https://console.aws.amazon.com/cloudwatch/home#logsV2>)