



**UNIFOR - UNIVERSIDADE DE FORTALEZA**  
**CURSO: ESPECIALIZAÇÃO EM ENGENHARIA DE DADOS**

**ATIVIDADE PRÁTICA: GUIA DE IMPLEMENTAÇÃO DE ENGENHARIA DE**  
**DADOS: PIPELINE DE ALERTA DE CHUVAS**

Aluno(s): José Wendemberg Henrique Lima

Professor: Marcondes Alexandre

Fortaleza - CE  
2025

## 1. Introdução

O presente trabalho tem como finalidade apresentar um guia detalhado para implementação de um projeto de engenharia de dados, aplicando serviços da Amazon Web Services (AWS) para ingestão, processamento e envio de dados meteorológicos em tempo real. O setor escolhido é meteorologia aplicada, pela relevância de informações precisas para setores críticos, como agricultura, logística, defesa civil e gestão urbana. O projeto foi desenvolvido para consumir dados de uma API externa (Tomorrow.io), processá-los de forma escalável utilizando AWS Lambda e Amazon Kinesis, armazenar e distribuir os resultados em tempo real e disparar alertas automáticos via Amazon SNS. Esta arquitetura permite demonstrar as principais etapas de um pipeline moderno de engenharia de dados e reforça boas práticas de segurança e governança.

## 2. Setor e Justificativa

O setor de meteorologia aplicada enfrenta desafios significativos no tratamento de dados. As informações meteorológicas são geradas em grande volume, alta frequência e de diversas fontes, exigindo pipelines escaláveis e altamente disponíveis. Outro ponto crítico é a necessidade de respostas rápidas para emissão de alertas (chuvas intensas, ventos fortes, riscos de enchentes), o que torna o processamento em tempo real indispensável. A escolha do setor se deve ao impacto direto dos dados meteorológicos em decisões estratégicas, como planejamento agrícola, logística de transporte e ações preventivas de defesa civil. O projeto demonstra como uma arquitetura de dados em nuvem pode atender a essas necessidades de forma eficiente e segura.

## 3. Definição do Problema

O problema abordado é a necessidade de processar dados meteorológicos em tempo real e gerar alertas automáticos para usuários finais e gestores. Com dados atualizados de APIs externas, é possível alimentar sistemas analíticos, dashboards e mecanismos de alerta, otimizando decisões e prevenindo riscos. O objetivo do projeto é criar um pipeline que consuma dados meteorológicos externos, armazene e processe fluxos de dados em tempo real, dispare alertas automáticos por SMS e e-mail e seja seguro, escalável e economicamente viável.

## 4. Arquitetura e Coleta de Dados

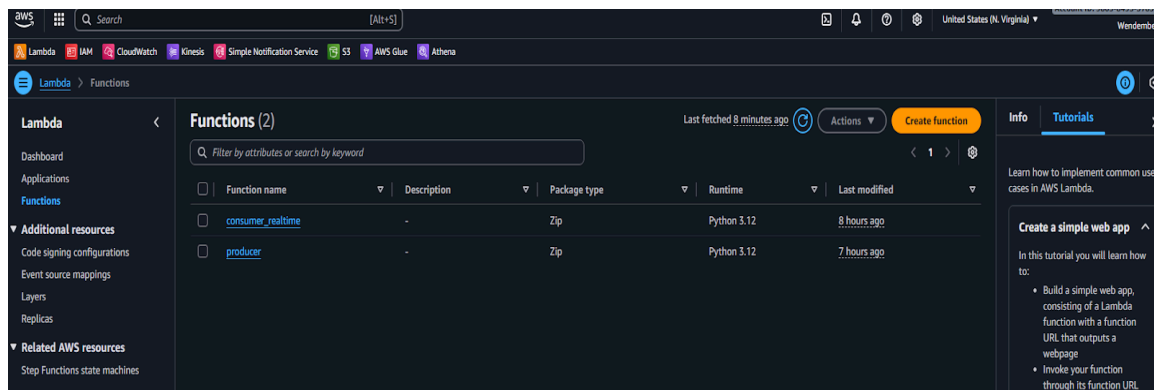
Os dados meteorológicos são coletados por meio da API Tomorrow.io, que fornece informações atualizadas sobre temperatura, precipitação, vento e outros indicadores climáticos. Uma Lambda Producer faz a chamada à API, processa o payload e envia para um Kinesis Data Stream (broker). Passos técnicos da coleta:

1. API Tomorrow.io: Fonte de dados meteorológicos consumida pela solução.

2. IAM Role (producer\_iam): Papel de segurança responsável por conceder permissões de execução à função Lambda produtora.
3. CloudWatch (producer\_event): Serviço de monitoramento utilizado para agendamento da execução da função Lambda produtora.
4. Lambda (producer): Função responsável por coletar os dados da API e enviá-los ao fluxo de ingestão.
5. Kinesis Data Stream (broker): Canal de ingestão em tempo real utilizado para intermediar a comunicação entre produtor e consumidor.
6. IAM Role (consumerrealtime\_iam): Papel de segurança associado ao consumo de dados em tempo real. Lambda (consumer\_realtime): Função que processa os dados em tempo real provenientes do Kinesis.
7. SNS (snsalerta): Serviço de notificação configurado para envio de alertas por SMS e e-mail com base nos dados processados.

## 5. Armazenamento e Processamento

A ingestão em tempo real é feita pelo Amazon Kinesis, que atua como broker para distribuir dados para consumidores. Um Lambda Consumer (consumer\_realtime) processa os dados e dispara eventos para o Amazon SNS.

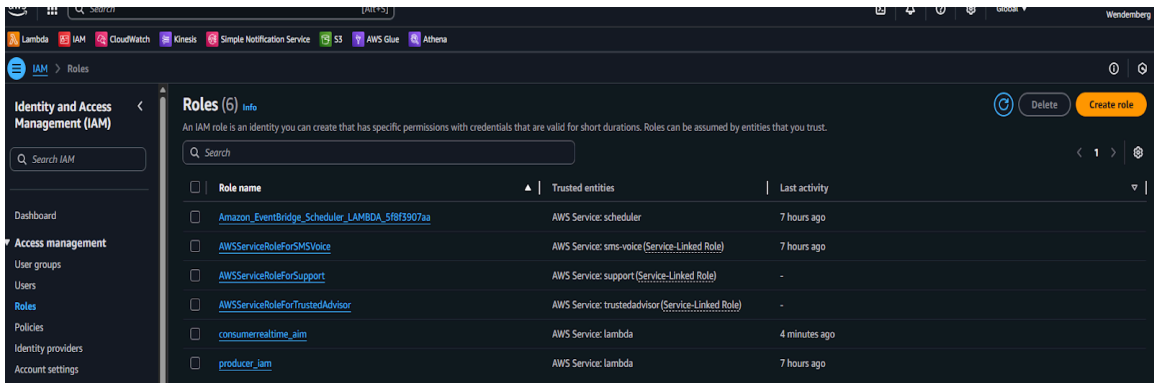


Pipeline resumido: API Tomorrow.io → Lambda Producer → Kinesis Broker → Lambda Consumer → SNS → SMS/Email. Essa abordagem elimina a necessidade de armazenamento intermediário em banco de dados para o streaming, mas nada impede que uma camada analítica (como um Data Lake no S3 ou BigQuery) seja integrada posteriormente para análises históricas.

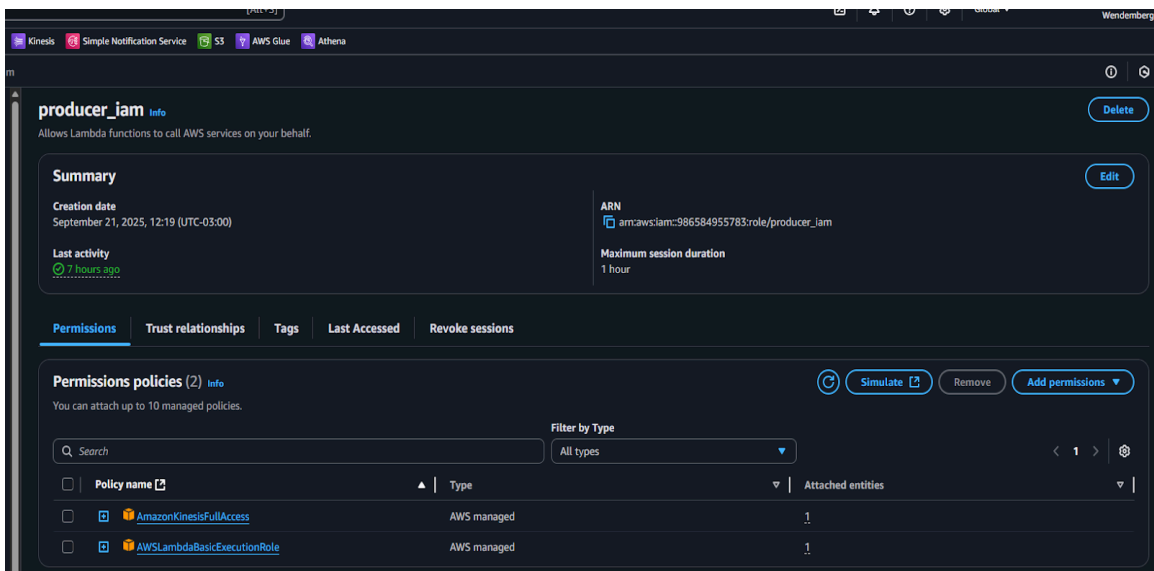
## 6. Perfis de Acesso (IAM Roles)

Para garantir segurança e separação de funções, foram criadas duas IAM Roles distintas:

producer\_iam

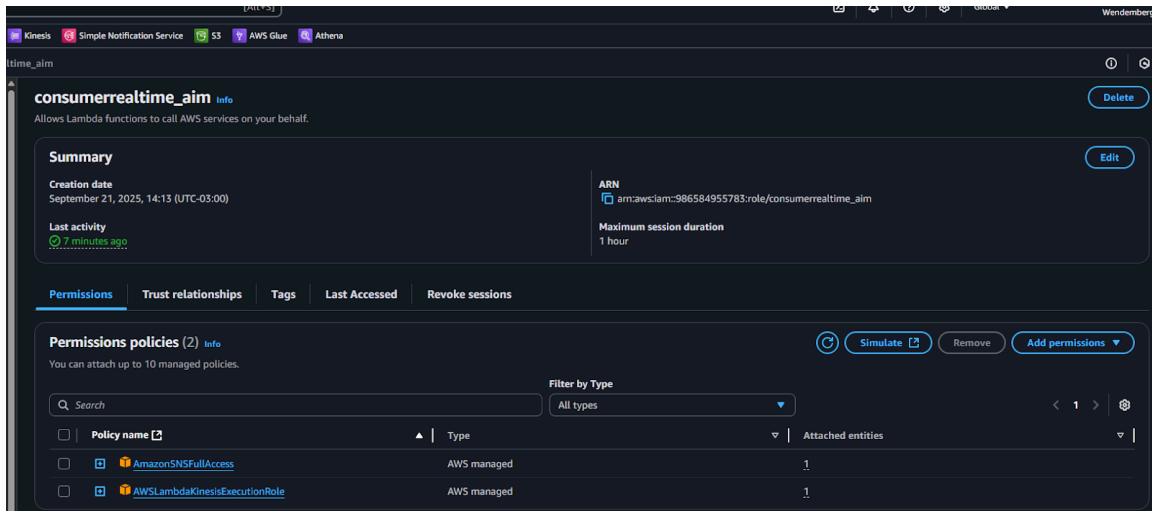


- Políticas: AmazonKinesisFullAccess e AWSLambdaBasicExecutionRole.



Responsável por permitir que a Lambda Producer grave dados no Kinesis e envie logs ao CloudWatch.

consumerrealtime\_iam



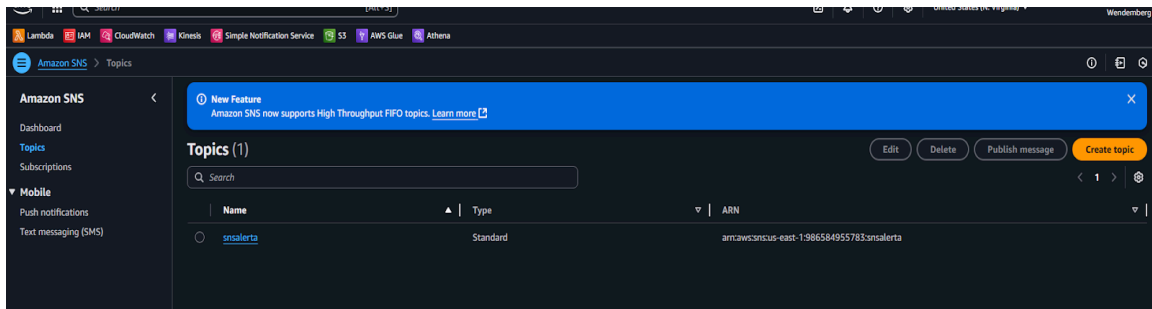
- Políticas: AmazonSNSFullAccess e AWSLambdaKinesisExecutionRole.

Responsável por permitir que a Lambda Consumer leia dados do Kinesis e publique mensagens no SNS.

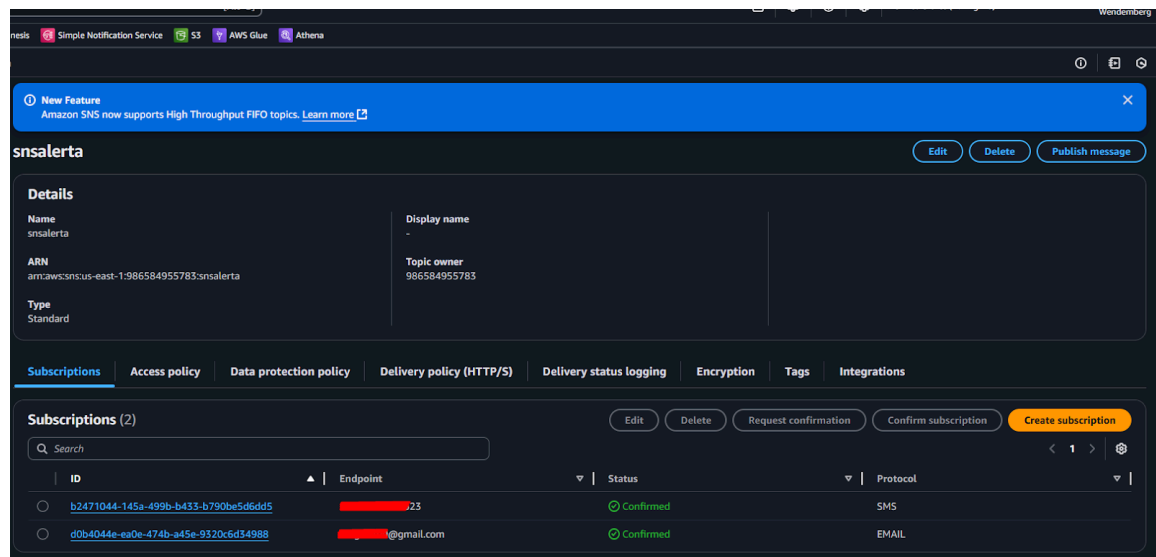
Essa separação reduz a superfície de ataque e segue o princípio do privilégio mínimo.

## 7. Alertas e Comunicação

Para disseminar informações em tempo real, foi criado o SNS Topic “snsalerta” com assinaturas para SMS e e-mail.



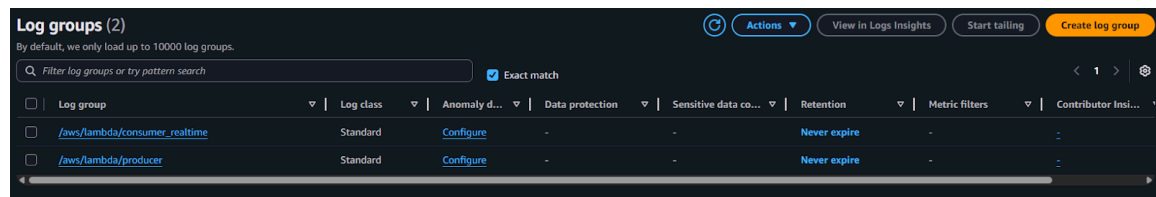
Cada novo dado processado pela Lambda Consumer gera um evento que dispara uma notificação para os inscritos.



O uso do Amazon SNS simplifica o envio de mensagens para múltiplos destinos, garantindo escalabilidade e alta disponibilidade para comunicações críticas.

## 8. Segurança e Governança dos Dados

O projeto considera as melhores práticas de segurança: criação de roles IAM específicas para cada função; uso do princípio do privilégio mínimo para políticas; monitoramento e logs centralizados no CloudWatch;



possibilidade de criptografia em trânsito e repouso (TLS/SSE); segregação de ambientes (dev, prod) e controle de versões.

Além disso, respeita diretrizes de LGPD e boas práticas de governança, garantindo que dados pessoais não sejam expostos.

## 9. Conclusão Real Time

O pipeline desenvolvido demonstra como integrar dados externos, processá-los em tempo real e enviar alertas automáticos usando serviços nativos da AWS. Ele é escalável, seguro e adaptável para diversos setores que necessitam de ingestão e resposta imediata a dados. O projeto pode ser expandido para incluir dashboards em tempo real, integração com bancos relacionais ou data lakes, machine learning para previsões avançadas e automações para gestão de custos.

## 10. Processamento em Batch (AWS Glue)

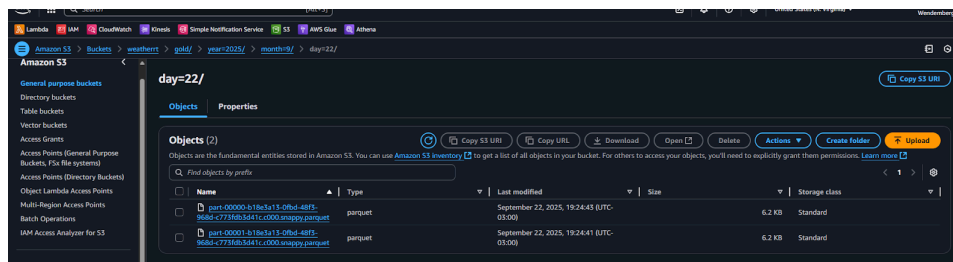
O componente de Batch é responsável pelo armazenamento, organização e disponibilização dos dados em lotes, permitindo análises históricas e consultas analíticas. Os principais elementos são:

1. Lambda (consumer\_batch): Função responsável por consumir dados e armazená-los no bucket S3.
2. S3 (raw): Camada de dados brutos (raw layer) para armazenamento inicial.
3. Crawler (raw\_crawler): Serviço de detecção de esquema aplicado sobre os dados brutos.
5. Catalog (raw\_db): Catálogo de metadados construído a partir da camada raw, utilizado pelo Athena.
6. Glue (weather\_job): Job de transformação dos dados (ETL) que gera a camada tratada.
7. IAM Role (etl\_role): Papel de segurança associado às execuções do Glue.
8. S3 (gold): Camada de dados tratados (gold layer), pronta para análises.
9. Crawler (gold\_crawler): Serviço de detecção de esquema aplicado sobre a camada gold.
10. Catalog (gold\_db): Catálogo de metadados para a camada tratada.
11. Athena: Serviço de consultas SQL aplicado aos dados armazenados no S3.
12. IAM Role (consumerbatch\_iam): Papel de segurança associado às execuções do componente batch.

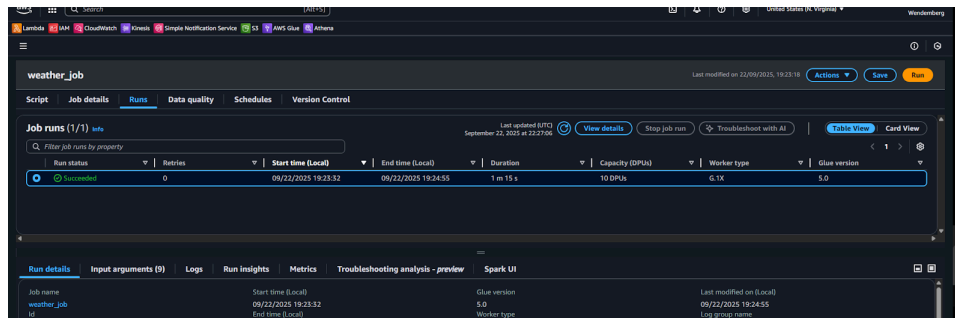
A arquitetura em batch utiliza como ponto central o Amazon S3 para armazenamento dos dados em múltiplas camadas:

- Raw: armazenamento de dados brutos no formato JSON.
- Gold: dados tratados e otimizados no formato Parquet, particionados por ano/mês/dia.

Os dados são organizados em pastas hierárquicas no bucket S3 (ex: s3://weatherrt/gold/year=2025/month=9/day=22/).



O AWS Glue é utilizado para processamento em batch, onde jobs são responsáveis por transformar os dados da camada raw para a camada gold. Esses jobs aplicam transformações ETL (Extract, Transform, Load), convertendo os dados para formatos otimizados e adicionando particionamento para melhorar o desempenho em consultas. A figura abaixo mostra a execução bem-sucedida de um Glue Job.



The screenshot displays the AWS Glue console interface for a job named 'weather\_job'. The 'Runs' tab is selected, showing a table of job runs. The first run is in a 'Succeeded' state. Below the table, the 'Run details' section provides specific information about the job's execution.

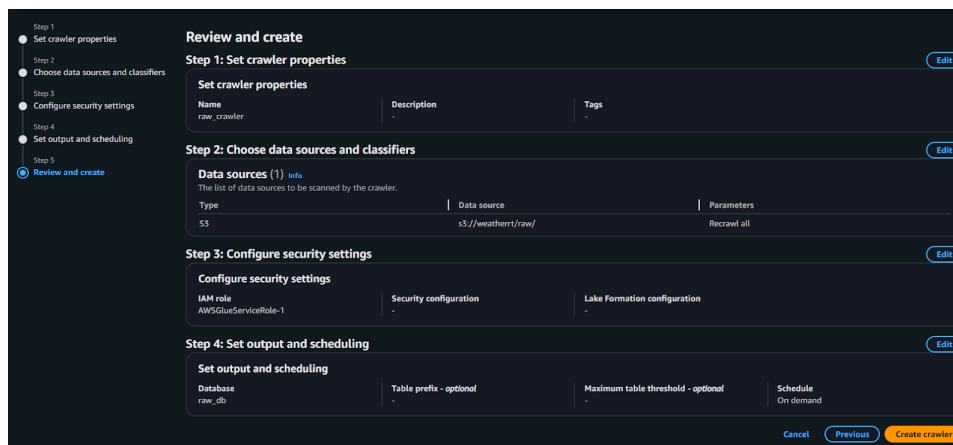
Run status	Retries	Start time (Local)	End time (Local)	Duration	Capacity (DPUh)	Worker type	Glue version
Succeeded	0	09/22/2025 19:28:32	09/22/2025 19:24:55	1 m 15 s	10 DPUh	G.1X	5.0

Job name	Start time (Local)	Glue version	Last modified on (Local)
weather_job	09/22/2025 19:28:32	5.0	09/22/2025 19:24:55

## 11. Catálogo de Dados (AWS Glue Crawler)

O AWS Glue Crawler é utilizado para varrer as pastas no S3 e criar tabelas automaticamente no Data Catalog. Isso permite que os dados armazenados em diferentes camadas sejam consultados diretamente pelo Amazon Athena.



The screenshot shows the 'Review and create' step of the AWS Glue Crawler setup wizard. It is divided into four sections: Step 1: Set crawler properties, Step 2: Choose data sources and classifiers, Step 3: Configure security settings, and Step 4: Set output and scheduling. Each section contains configuration details for the crawler.

Name	Description	Tags
raw_crawler	-	-

Type	Data source	Parameters
S3	s3://weather1/raw/	Recrawl all

IAM role	Security configuration	Lake Formation configuration
AWSGlueServiceRole-1	-	-

Database	Table prefix - optional	Maximum table threshold - optional	Schedule
raw_db	-	-	On demand

## 12. Estrutura das Tabelas

Os crawlers criaram tabelas representando os dados meteorológicos com diferentes níveis de processamento. A camada raw possui schema baseado em JSON, enquanto a camada gold possui schema otimizado em Parquet.

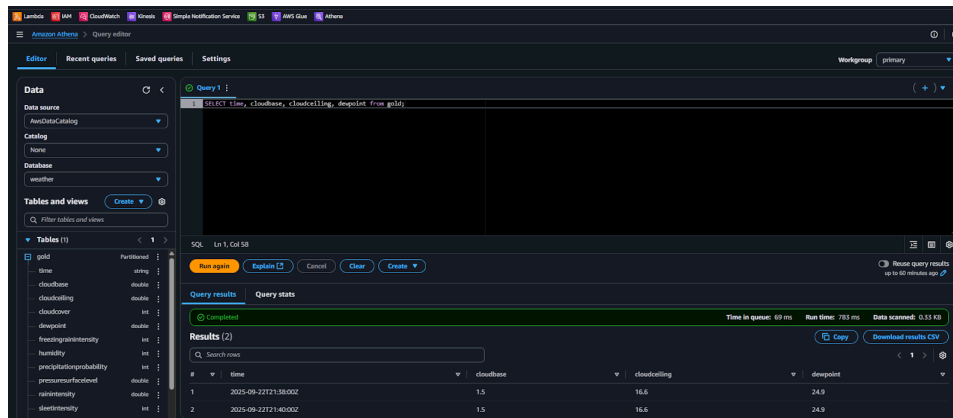


Table details																																		
Name	raw	Classification	JSON	Deprecated																														
Database	raw_db	Location	s3://weatherfr/raw/	Column statistics																														
Description	-	Connection	-	No statistics																														
Last updated September 22, 2025 at 22:03:14																																		
Advanced properties																																		
<div>Schema</div> <div>Partitions</div> <div>Indexes</div> <div>Column statistics - new</div>																																		
<div>Schema (5)</div> <div>View and manage the table schema.</div> <div> <input type="text" value="Filter schemas"/> <div>Edit schema as JSON</div> <div>Edit schema</div> </div> <table> <tr> <th>#</th><th>Column name</th><th>Data type</th><th>Partition key</th><th>Comment</th></tr> <tr> <td>1</td><td>data</td><td>struct</td><td>-</td><td>-</td></tr> <tr> <td>2</td><td>location</td><td>struct</td><td>-</td><td>-</td></tr> <tr> <td>3</td><td>year</td><td>string</td><td>Partition (0)</td><td>-</td></tr> <tr> <td>4</td><td>month</td><td>string</td><td>Partition (1)</td><td>-</td></tr> <tr> <td>5</td><td>day</td><td>string</td><td>Partition (2)</td><td>-</td></tr> </table>					#	Column name	Data type	Partition key	Comment	1	data	struct	-	-	2	location	struct	-	-	3	year	string	Partition (0)	-	4	month	string	Partition (1)	-	5	day	string	Partition (2)	-
#	Column name	Data type	Partition key	Comment																														
1	data	struct	-	-																														
2	location	struct	-	-																														
3	year	string	Partition (0)	-																														
4	month	string	Partition (1)	-																														
5	day	string	Partition (2)	-																														

Schema (26)																																																																																																													
View and manage the table schema.																																																																																																													
<div> <input type="text" value="Filter schemas"/> <div>Edit schema as JSON</div> <div>Edit schema</div> </div> <table> <tr> <th>#</th><th>Column name</th><th>Data type</th><th>Partition key</th><th>Comment</th></tr> <tr><td>1</td><td>time</td><td>string</td><td>-</td><td>-</td></tr> <tr><td>2</td><td>cloudbase</td><td>double</td><td>-</td><td>-</td></tr> <tr><td>3</td><td>cloudceiling</td><td>double</td><td>-</td><td>-</td></tr> <tr><td>4</td><td>cloudcover</td><td>int</td><td>-</td><td>-</td></tr> <tr><td>5</td><td>dewpoint</td><td>double</td><td>-</td><td>-</td></tr> <tr><td>6</td><td>freezingrainintensity</td><td>int</td><td>-</td><td>-</td></tr> <tr><td>7</td><td>humidity</td><td>int</td><td>-</td><td>-</td></tr> <tr><td>8</td><td>precipitationprobability</td><td>int</td><td>-</td><td>-</td></tr> <tr><td>9</td><td>pressureatsealevel</td><td>double</td><td>-</td><td>-</td></tr> <tr><td>10</td><td>rainintensity</td><td>double</td><td>-</td><td>-</td></tr> <tr><td>11</td><td>sleetintensity</td><td>int</td><td>-</td><td>-</td></tr> <tr><td>12</td><td>snowintensity</td><td>int</td><td>-</td><td>-</td></tr> <tr><td>13</td><td>temperature</td><td>double</td><td>-</td><td>-</td></tr> <tr><td>14</td><td>temperatureapparent</td><td>double</td><td>-</td><td>-</td></tr> <tr><td>15</td><td>uvhealthconcern</td><td>int</td><td>-</td><td>-</td></tr> <tr><td>16</td><td>visindex</td><td>int</td><td>-</td><td>-</td></tr> <tr><td>17</td><td>visibility</td><td>int</td><td>-</td><td>-</td></tr> <tr><td>18</td><td>weathercode</td><td>int</td><td>-</td><td>-</td></tr> <tr><td>19</td><td>winddirection</td><td>int</td><td>-</td><td>-</td></tr> <tr><td>20</td><td>windgust</td><td>double</td><td>-</td><td>-</td></tr> </table>					#	Column name	Data type	Partition key	Comment	1	time	string	-	-	2	cloudbase	double	-	-	3	cloudceiling	double	-	-	4	cloudcover	int	-	-	5	dewpoint	double	-	-	6	freezingrainintensity	int	-	-	7	humidity	int	-	-	8	precipitationprobability	int	-	-	9	pressureatsealevel	double	-	-	10	rainintensity	double	-	-	11	sleetintensity	int	-	-	12	snowintensity	int	-	-	13	temperature	double	-	-	14	temperatureapparent	double	-	-	15	uvhealthconcern	int	-	-	16	visindex	int	-	-	17	visibility	int	-	-	18	weathercode	int	-	-	19	winddirection	int	-	-	20	windgust	double	-	-
#	Column name	Data type	Partition key	Comment																																																																																																									
1	time	string	-	-																																																																																																									
2	cloudbase	double	-	-																																																																																																									
3	cloudceiling	double	-	-																																																																																																									
4	cloudcover	int	-	-																																																																																																									
5	dewpoint	double	-	-																																																																																																									
6	freezingrainintensity	int	-	-																																																																																																									
7	humidity	int	-	-																																																																																																									
8	precipitationprobability	int	-	-																																																																																																									
9	pressureatsealevel	double	-	-																																																																																																									
10	rainintensity	double	-	-																																																																																																									
11	sleetintensity	int	-	-																																																																																																									
12	snowintensity	int	-	-																																																																																																									
13	temperature	double	-	-																																																																																																									
14	temperatureapparent	double	-	-																																																																																																									
15	uvhealthconcern	int	-	-																																																																																																									
16	visindex	int	-	-																																																																																																									
17	visibility	int	-	-																																																																																																									
18	weathercode	int	-	-																																																																																																									
19	winddirection	int	-	-																																																																																																									
20	windgust	double	-	-																																																																																																									

### 13. Consultas no Athena

Após os dados estarem organizados e catalogados, o Amazon Athena é utilizado para executar queries SQL diretamente sobre os arquivos no S3. Isso elimina a necessidade de movimentação dos dados para um banco de dados relacional tradicional, garantindo maior eficiência e escalabilidade.



## 14. Segurança e Governança dos Dados

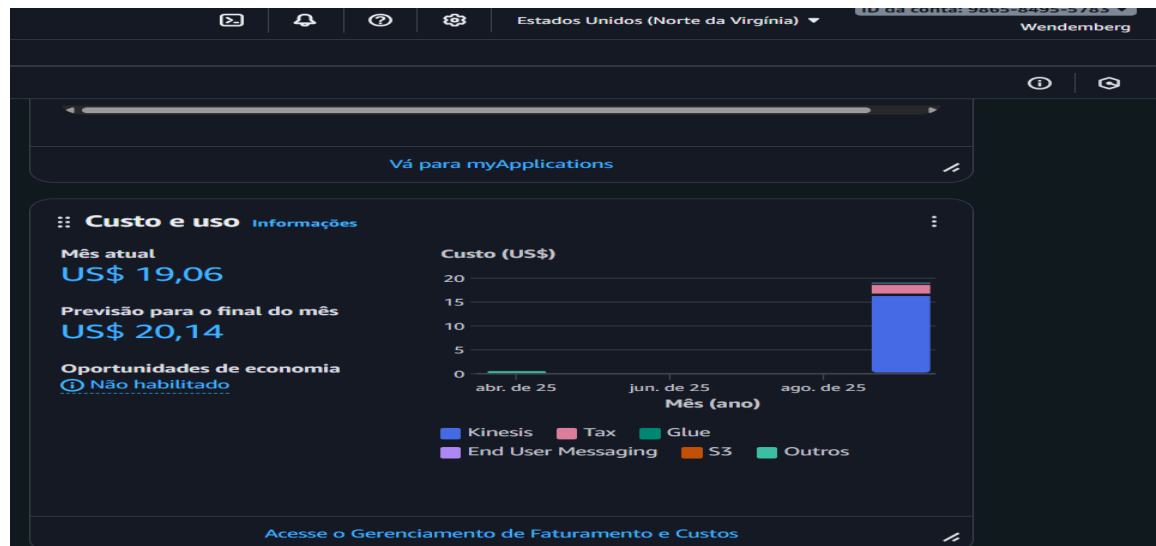
Assim como no processo em streaming, foram aplicadas boas práticas de segurança:

- Criação de roles IAM específicas para Glue e Athena.
- Uso do princípio do privilégio mínimo.
- Monitoramento e logs centralizados.
- Possibilidade de criptografia em trânsito e repouso (TLS/SSE).
- Segregação de ambientes (dev, prod).

Além disso, o pipeline respeita diretrizes da LGPD, evitando exposição de dados sensíveis.

## 15. Conclusão

O processo em batch implementado demonstra como organizar dados meteorológicos em diferentes camadas dentro de um Data Lake baseado em Amazon S3, utilizando AWS Glue para transformação e Amazon Athena para análise. A arquitetura oferece escalabilidade, custo-benefício e flexibilidade para integração com dashboards e modelos preditivos. O projeto rodou do dia 20 até dia 30.



## 16. Referências

- AWS Documentation: Glue, Athena, S3, IAM, : Lambda, Kinesis, SNS, IAM Roles
- Tomorrow.io API Documentation.
- LGPD – Lei Geral de Proteção de Dados (Lei nº 13.709/2018).
- <https://www.tomorrow.io/>