



**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

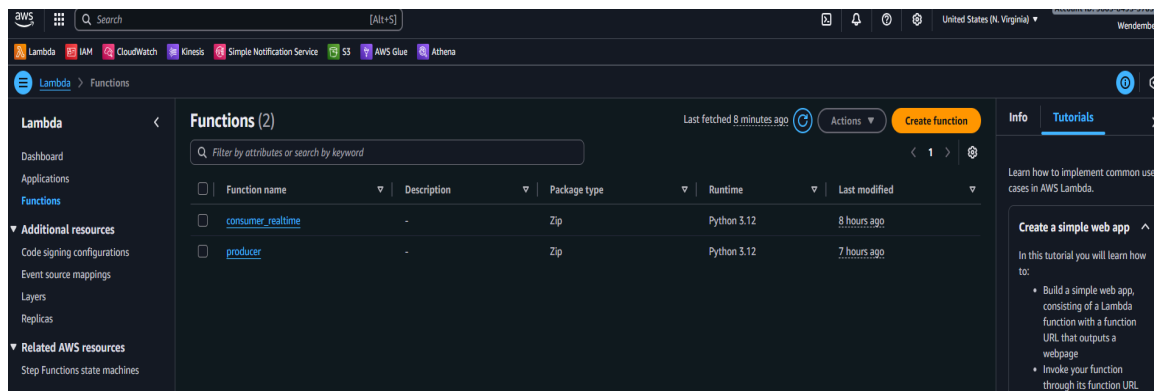
Os dados meteorológicos são coletados por meio da API Tomorrow.io, que fornece informações atualizadas sobre temperatura, precipitação, vento e outros indicadores climáticos. Uma Lambda Producer faz a chamada à API, processa o payload e envia para um Kinesis Data Stream (broker). Passos técnicos da coleta:

1. Criação de chave de API Tomorrow.io.
2. Desenvolvimento da função Lambda “producer” para consumir os dados.

3. Criação do Kinesis Data Stream “broker” para ingestão em tempo real.
4. Configuração do CloudWatch para agendar execuções e monitorar logs.

## 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.

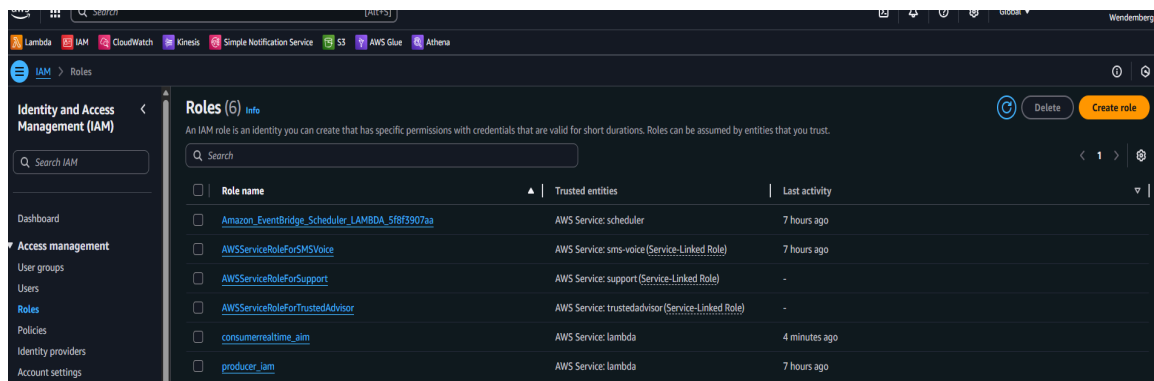


Pipeline resumido: API Tomorrow.io → Lambda Producer → Kinesis Broker → Lambda Consumer → SNS → SMS/Email. Essa abordagem elimina a necessidade de armazenamento intermediário em banco de dados para o streaming, mas nada impede que uma camada analítica (como um Data Lake no S3 ou BigQuery) seja integrada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

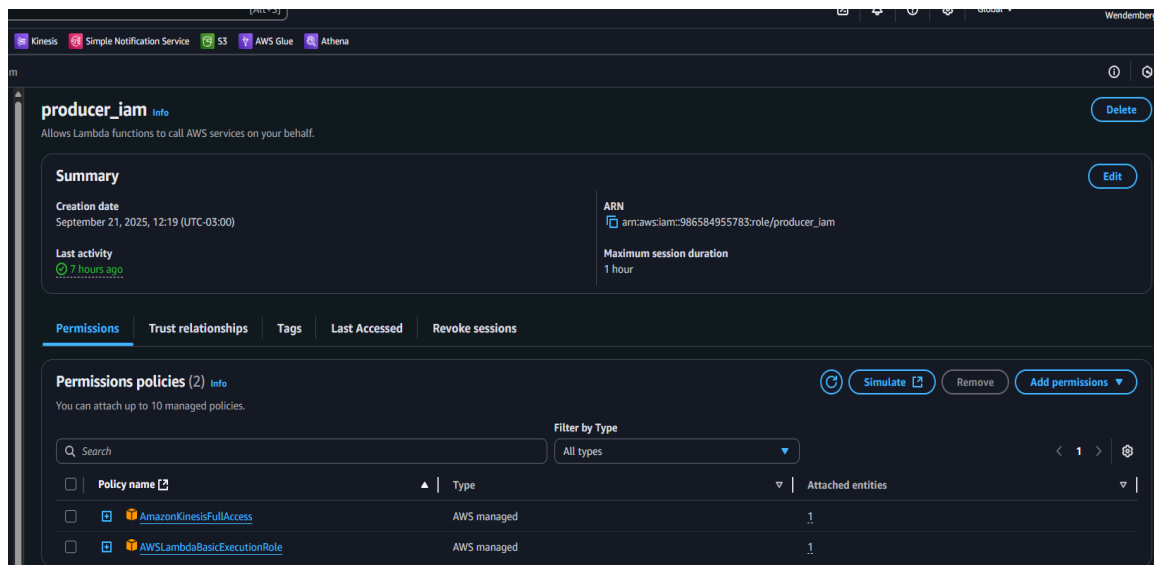
## 6. Perfis de Acesso (IAM Roles)

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

producer\_iam

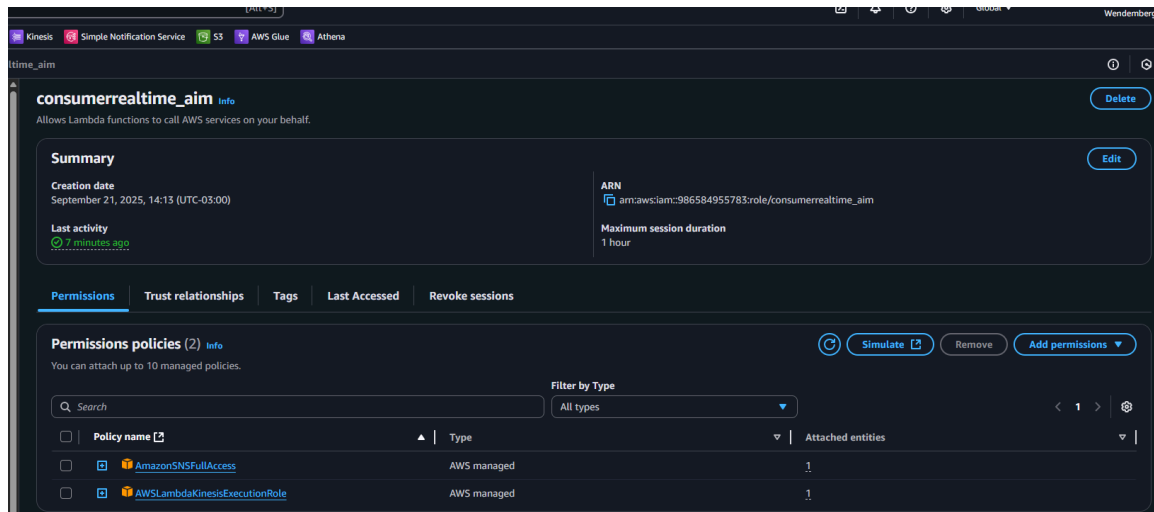


- Políticas: AmazonKinesisFullAccess e AWSLambdaBasicExecutionRole.



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

consumerrealtime\_iam



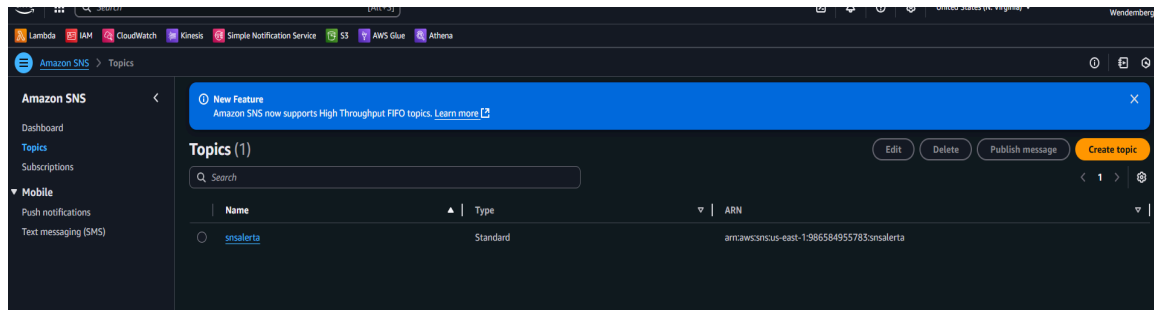
- Políticas: AmazonSNSFullAccess e AWSLambdaKinesisExecutionRole.

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

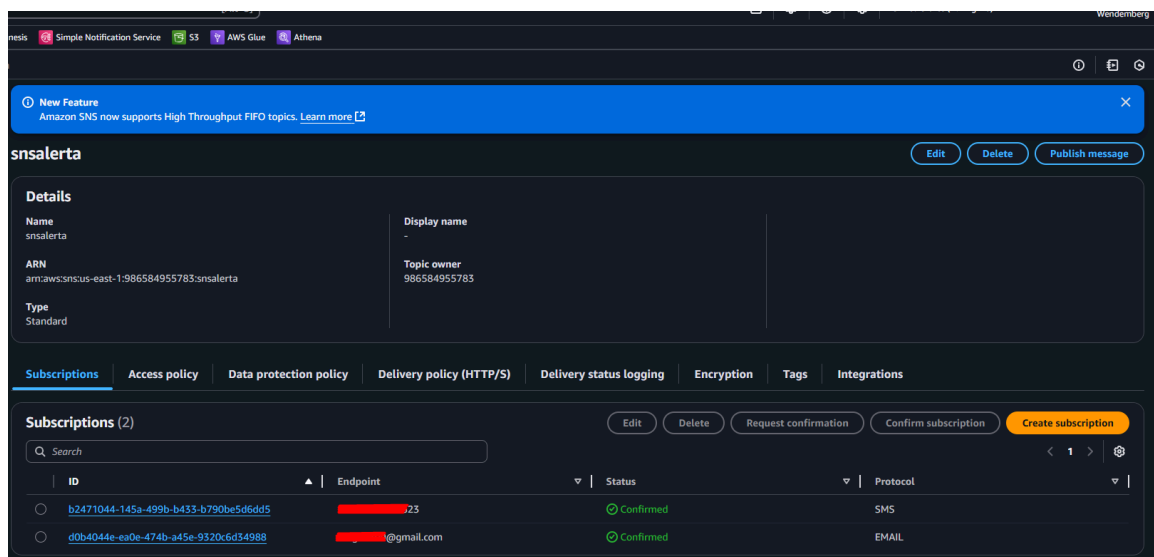
Essa separação reduz a superfície de ataque e segue o princípio do privilégio mínimo.

## 7. Alertas e Comunicação

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



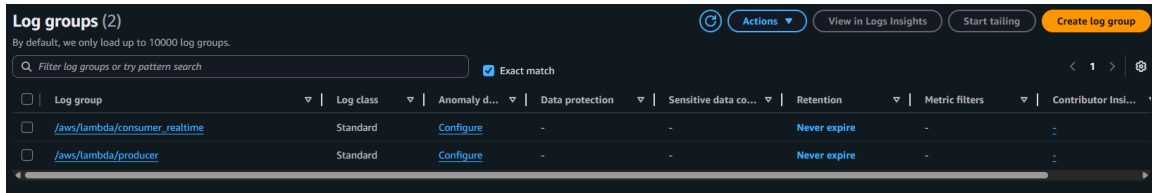
Cada novo dado processado pela Lambda Consumer gera um evento que dispara uma notificação para os inscritos.



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

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. Referências

- AWS Documentation: 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/>
- [https://github.com/Wendemberg/Aws\\_pipeline](https://github.com/Wendemberg/Aws_pipeline)