Semana do Desenvolvedor – Dia 3

Aula 3: Processamento Central de Pedidos e Persistência

Documentação do Laboratório: Arquitetura de Processamento de Pedidos e Arquivos na AWS

(Duração Estimada: 160 minutos)

1. Visão Geral e Objetivos da Aula

Nas aulas anteriores, estabelecemos dois canais de entrada para pedidos (API e S3) que, após validação, resultam na publicação de um evento NovoPedidoValidado no nosso Custom event bus no Amazon EventBridge.

Nesta terceira aula, vamos construir a lógica que consome esses eventos. Configuraremos uma regra no EventBridge para capturar os eventos de NovoPedidoValidado. Essa regra direcionará os eventos para uma nova fila SQS Standard, que servirá como um buffer para o processamento principal. Uma nova função Lambda será acionada por esta fila, simulará a lógica de "processamento do pedido" (ex: verificação de inventário, cálculo de frete, etc. - que simplificaremos) e, finalmente, persistirá os detalhes e o status do pedido processado em uma nova tabela principal do Amazon DynamoDB.

Ao final desta aula, você terá:

- Uma regra no EventBridge para rotear eventos de novos pedidos validados.
- Uma fila SQS Standard para desacoplar o processamento principal dos pedidos.
- Uma função Lambda para executar a lógica de processamento central do pedido.
- Uma tabela DynamoDB principal para armazenar o estado e os detalhes dos pedidos processados.
- A conexão efetiva entre a validação do pedido e sua persistência final.

Recursos a Serem Criados Nesta Aula:

- IAM Role: lambda-processa-pedidos-role-seu-nome
- Amazon SQS DLQ (Standard): pedidos-pendentes-dlq-seu-nome
- Amazon SQS Queue (Standard): pedidos-pendentes-queue-seu-nome

- AWS Lambda Function: processa-pedidos-lambda-seu-nome
- Amazon DynamoDB Table: pedidos-db-seu-nome (Tabela principal de pedidos)
- Amazon EventBridge Rule: novo-pedido-validado-rule-seu-nome

Lembrete Importante: Substitua seu-nome em todos os nomes de recursos. Trabalhe consistentemente na mesma região AWS.

---

2. Configuração de Permissões (IAM Role)

Criaremos uma IAM Role para a Lambda que realizará o processamento central dos pedidos.

2.1. Criar Role para a Lambda de Processamento de Pedidos (lambda-processa-pedidos-role-seu-nome)

1. Acesse o serviço **IAM** no Console AWS.
2. No menu à esquerda, clique em **"Roles"** e depois em **"Create role"**.
3. **Trusted entity type:** Selecione **"AWS service"**.
4. **Use case:** Selecione **"Lambda"** e clique em **"Next"**.
5. Na página **"Add permissions"**:
    - Procure pela política AWSLambdaBasicExecutionRole e marque-a.
    - Clique em **"Next"**.
6. **Role details**:
    - **Role name:** lambda-processa-pedidos-role-seu-nome
    - **Description:** (Opcional) Ex: "Role para Lambda de processamento central de pedidos".
7. Clique em **"Create role"**.
    - ***Nota:*** *Após criar a SQS e a DynamoDB, voltaremos para adicionar as permissões específicas.*

---

3. Configuração do Enfileiramento de Pedidos Pendentes (Amazon SQS Standard)

Esta fila SQS receberá os eventos de NovoPedidoValidado do EventBridge e desacoplará a Lambda de processamento.

3.1. Criar a Fila de Mensagens Mortas (DLQ) (pedidos-pendentes-dlq-seu-nome)

1. Acesse o serviço **Amazon SQS**.
2. Clique em **"Create queue"**.
3. **Type:** Selecione **"Standard"**.
4. **Name:** pedidos-pendentes-dlq-seu-nome.
5. **Visibility timeout:** Mantenha o padrão de 30 segundos (ou ajuste se o timeout da Lambda de processamento for maior, mas para este exemplo, 30s é suficiente para o timeout da Lambda que definiremos).
6. Clique em **"Create queue"**.
7. **Anote o ARN** desta DLQ.

3.2. Criar a Fila Principal de Pedidos Pendentes (pedidos-pendentes-queue-seu-nome)

1. Ainda no **SQS > Queues**, clique em **"Create queue"**.
2. **Type:** Selecione **"Standard"**.
3. **Name:** pedidos-pendentes-queue-seu-nome.
4. **Visibility timeout:** Altere para 70 **Seconds**. (Assumindo que o timeout da Lambda processa-pedidos-lambda-seu-nome será configurado para 60 segundos, como fizemos na Aula 2).
5. Na seção **"Dead-letter queue (DLQ)"**:
    - Marque **"Enabled"**.
    - Em **Choose queue,** selecione o ARN da fila que você criou no passo anterior**: pedidos-pendentes-dlq-seu-nome**.
    - **Maximum receives:** Defina como 3.
6. Clique em **"Create queue"**.
7. **Anote o ARN e a URL** desta fila principal.

---

4. Configuração do Banco de Dados Principal de Pedidos (Amazon DynamoDB)

Esta tabela DynamoDB armazenará as informações finais dos pedidos processados.

4.1. Criar a Tabela DynamoDB Principal (pedidos-db-seu-nome)

1. Acesse o serviço **Amazon DynamoDB > Tables**.
2. Clique em **"Create table"**.
3. **Table details:**
    - **Table name:** pedidos-db-seu-nome
    - **Partition key:** pedidoId.
    - **Partition key type:** "String".
4. Mantenha as demais configurações padrão (**Table settings > Default settings**).
5. Clique em **"Create table"**. A criação pode levar alguns instantes.
6. Selecione a tabela recém-criada e **anote o ARN** desta tabela DynamoDB.

---

5. Atualizar Permissões da Role da Lambda de Processamento de Pedidos

Agora que a SQS e a DynamoDB estão criadas, vamos conceder à lambda-processa-pedidos-role-seu-nome as permissões necessárias.

1. Volte ao serviço **IAM**.
2. Clique em **"Roles"** e encontre a role lambda-processa-pedidos-role-seu-nome. Clique nela.
3. Na aba **"Permissions"**, clique em **"Add permissions"** e selecione **"Create inline policy"**.
4. Selecione a aba **JSON**.
5. Remova a política existente e cole a política abaixo, modificando as seguintes informações:

```
{
  "Version": "2012-10-17",
  "Statement": [
   {
    "Effect": "Allow",
    "Action": [
     "sqs:ReceiveMessage",
     "sqs:DeleteMessage",
     "sqs:GetQueueAttributes"
    ],
    "Resource": "COLE AQUI O ARN DA SUA FILA pedidos-pendentes-queue-seu-nome"
   },
   {
    "Effect": "Allow",
    "Action": [
     "dynamodb:PutItem",
     "dynamodb:UpdateItem",
     "dynamodb:GetItem"
    ],
    "Resource": "COLE AQUI O ARN DA SUA TABELA pedidos-db-seu-nome"
   }
  ]
}
```

6. Clique em **"Next"**.
7. Em **"Review and create"**:
   - **Policy name:** SQSReadDynamoWritePedidosDB-seu-nome.
   - Clique em **"Create policy"**.

---

6. Implementação da Lógica de Processamento de Pedidos (AWS Lambda)

Esta função Lambda será acionada por mensagens da fila pedidos-pendentes-queue-seu-nome e persistirá os dados no DynamoDB.

6.1. Criar a Função Lambda (processa-pedidos-lambda-seu-nome)

1. Acesse o serviço **AWS Lambda**.
2. Clique em **"Create function"** > **"Author from scratch"**.
3. **Function name:** processa-pedidos-lambda-seu-nome.
4. **Runtime:** Python 3.12.
5. Expanda a seção **“Change default execution role”**
6. Selecione **Use an existing role.**
7. **Existing role:** Use a role **lambda-processa-pedidos-role-seu-nome**.
8. Clique em **"Create function"**.

6.2. Configurar o Código, Variáveis de Ambiente e Trigger SQS

1. Na página da função, rola para baixo, até a aba **"Code"**, substitua lambda_function.py pelo código Python que você encontra, **clicando aqui**.
2. Clique em **"Deploy"**.
3. Vá para a aba **"Configuration"** > **"Environment variables"** > **"Edit"**.
4. Adicione a variável:
    - **Key:** DYNAMODB_TABLE_NAME
    - **Value:** pedidos-db-seu-nome
5. Clique em **"Save"**.
6. Ainda na aba **"Configuration"**, selecione **"General configuration"** > **"Edit"**.
    - Aumente o **Timeout** para 60 segundos.
    - Clique em **"Save"**.
7. Vá para **"Configuration"** > **"Triggers"** > **"Add trigger"**.
    - Fonte: **SQS**.
    - SQS queue: pedidos-pendentes-queue-seu-nome.
    - Batch size: 1.
    - Clique em **"Add"**.

---

7. Configuração da Regra no Amazon EventBridge

Criaremos uma regra no Custom event bus (criado na Aula 1) para capturar os eventos NovoPedidoValidado e direcioná-los para a fila pedidos-pendentes-queue-seu-nome.

7.1. Criar a Regra (novo-pedido-validado-rule-seu-nome)

1. Acesse o serviço **Amazon EventBridge**.
2. No menu à esquerda, clique em **"Rules"**.
3. Na lista suspensa "Event bus", selecione seu barramento de eventos customizado: pedidos-event-bus-seu-nome.
4. Clique em **"Create rule"**.
5. **Define rule detail:**
    - **Name:** novo-pedido-validado-rule-seu-nome
    - **Description:** (Opcional) Ex: "Regra para rotear novos pedidos validados para processamento".
    - **Event bus:** Deve estar pedidos-event-bus-seu-nome.
    - **Rule type:** Selecione **"Rule with an event pattern"**.
    - Clique em **"Next"**.
6. **Build event pattern:**
    - **Event source:** Selecione **"AWS events or EventBridge partner events"**.
    - Role para baixo até **"Event pattern"**.
    - **Creation method:** Selecione **"Custom patterns (JSON editor)"**.
    - No editor JSON, cole o seguinte padrão de evento. Certifique-se de que o valor de "source" corresponda exatamente ao que a Lambda validacao-pedidos-lambda-seu-nome (da Aula 1) está publicando:

```
{
  "source": ["lab.aula1.pedidos.validacao"],
  "detail-type": ["NovoPedidoValidado"]
}
```

- Clique em **"Next"**.

Observação: Esses valores vêm do código da sua função Lambda que publica o evento no EventBridge (a Lambda de “validação de pedidos” da Aula 1). Na chamada ao SDK da AWS (Events.putEvents), você define explicitamente:

- **source**: a string que identifica de onde o evento veio (por isso “lab.aula1.pedidos.validacao”).
- **detail-type**: o tipo de detalhe do evento, para categorizar (“NovoPedidoValidado”).

Ou seja, quando você escreveu o código Python da Lambda, algo como:

```
pythonCopiarEditarclient.put_events(
 Entries=[
  {
   'Source': 'lab.aula1.pedidos.validacao',
   'DetailType': 'NovoPedidoValidado',
   'Detail': json.dumps({...}),
   'EventBusName': os.environ['EVENT_BUS_NAME']
  }
 ]
)
```

Foi ali que nós “criamos” esses valores. É exatamente por isso que, ao montar a regra no EventBridge, é fundamental usar o mesmo **source** e **detail-type** que sua Lambda está enviando. Se você mudar para outro nome sem alterar o código de publicação, não haverá correspondência e nenhum evento será capturado.

7. **Select target(s):**
   - **Target types:** Selecione **"AWS service"**.
   - **Select a target:** Escolha **"SQS queue"** na lista suspensa.
   - **Queue:** Selecione sua fila pedidos-pendentes-queue-seu-nome na lista.
   - (Deixe "Additional settings" padrão).
   - Clique em **"Next"**.
8. **Configure tags - optional:** Pode pular esta etapa clicando em **"Next"**.

9. **Review and create:** Revise as configurações e clique em **"Create rule"**.

---

8. Teste do Fluxo Completo (API/S3 -> EventBridge -> SQS -> Lambda -> DynamoDB)

Agora vamos testar todo o pipeline, desde a entrada do pedido (via API ou via arquivo S3, já que ambos agora alimentam o mesmo EventBridge) até sua persistência final.

8.1. Enviar um Novo Pedido via API (Fluxo da Aula 1)

Prepare a Requisição

Você enviará uma requisição POST para a URL da sua API com um payload em formato JSON. Utilize o comando que você encontrará, **clicando aqui**, e altere os **valores abaixo:**

- **<INVOKE_URL>: URL de Invocação (Invoke URL) da sua API** que você anotou anteriormente. **Caminho:**
  - API Gateway > APIs > pedidos-api-seu-nome > Stages
- **pedidoId** e **clienteId**: personalize com os dados:
  - "**pedidoId**": "apiP001-seu-nome"
  - "**clienteId**": "clienteAPI-XYZ-seu-nome",
  - "**itens**": [{"item": "Produto X API", "qtd": 1}]
- **Exemplo com URL fictício de como deverá ficar:**

```
curl -X POST https://rtc9oh0au6.execute-api.us-east-1.amazonaws.com/dev/pedidos \
  -H "Content-Type: application/json" \
  -d '{
    "pedidoId": "apiP001-seu-nome",
    "clienteId": "clienteXYZ-seu-nome",
    "itens": [
      {
        "produto": "Produto X API",
        "quantidade": 1
      }
    ]
  }'
```

8.2. Fazer Upload de um Arquivo com Pedidos no S3 (Fluxo da Aula 2)

1. Baixe o arquivo **outro_arquivo_pedidos.json**, **clicando aqui**, com o seguinte conteúdo:

```
{
  "metadata_arquivo": {"data_geracao": "2024-03-16"},
  "lista_pedidos": [
    {
      "id_pedido_arquivo": "s3P003-seu-nome",
      "id_cliente_arquivo": "clienteS3-ABC",
      "itens_pedido_arquivo": [{"sku": "PROD-S3-001", "qtd": 10}]
    }
```

2. Após baixar, abra-o no seu editor de texto e modifique-o colocando o seu nome em: "s3P003-**seu-nome**".
3. **Salve.**
4. Faça o upload deste arquivo para o seu bucket S3 datalake-arquivos-seu-nome.

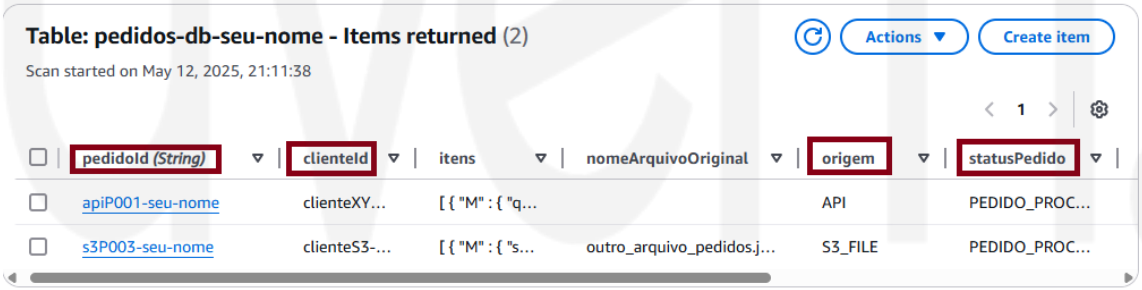
8.3. Verificar o Processamento e Persistência

Aguarde alguns momentos para que os pedidos passem por todo o fluxo.

1. **Logs da validacao-pedidos-lambda-seu-nome (Lambda da Aula 1):**
    - Verifique no **CloudWatch**. Você deverá ver logs para o pedido **apiP001-seu-nome** (da API) e para **s3P003-seu-nome** (do arquivo S3), ambos publicando um evento NovoPedidoValidado no EventBridge.
    - **Caminho:** CloudWatch > Logs > Log groups > /aws/lambda/validacao-s3-arquivos-lambda-seu-nome
    - **Você deve ver:**
        - "Evento SQS recebido".
        - "Processando pedido do SQS".
        - "Pedido X validado com sucesso".
        - "Evento publicado no EventBridge".
2. **EventBridge Rule Monitoring:**
    - No console do **EventBridge > Event buses**, selecione seu pedidos-event-bus-seu-nome.
    - Clique na regra novo-pedido-validado-rule-seu-nome.

- Na aba "Monitoring", observe as métricas como **Invocations** ou **TriggeredRules**. Você deverá ver um aumento correspondente aos eventos processados.

3. **Fila SQS de Pedidos Pendentes (pedidos-pendentes-queue-seu-nome):**
   - No SQS, observe esta fila. As mensagens (contendo os detalhes dos pedidos apiP001-seu-nome e s3P003-seu-nome). **É normal você chegar até aqui e não ver nenhuma mensagem no pool, pois ela é consumida rapidamente pelo seu lambda correspondente**.
4. **Logs da processa-pedidos-lambda-seu-nome (Lambda desta Aula 3):**
   - No CloudWatch, acesse os logs desta Lambda.
   - Você deverá ver logs indicando o recebimento das mensagens da SQS e o processamento dos pedidos apiP001-seu-nome e s3P003-seu-nome.
   - Procure por mensagens como: "Processando pedido: { ... 'pedidoId': 'apiP001-seu-nome' ... }" e "Pedido apiP001-seu-nome salvo/atualizado no DynamoDB com status PEDIDO_PROCESSADO.". Repita para o pedido S3.
5. **Tabela DynamoDB Principal (pedidos-db-seu-nome):**
   - No **DynamoDB > Tables**, selecione a tabela pedidos-db-seu-nome.
   - Clique em **"Explore table items"**.
   - Você deverá encontrar dois novos itens:
     - Um com **pedidoId** igual a **apiP001-seu-nome**, **statusPedido** igual a **PEDIDO_PROCESSADO**, e origem provavelmente API (ou o que sua Lambda da Aula 1 definir por padrão se não houver campo de origem).
     - Outro com **pedidoId** igual a **s3P003-seu-nome**, **statusPedido** igual a **PEDIDO_PROCESSADO**, e origem igual a **S3_FILE**.



- Verifique os outros atributos para garantir que os dados foram persistidos corretamente.

Conclusão da Aula 3

Excelente! Você conectou com sucesso o pipeline de validação de pedidos ao seu processamento central e persistência. Agora, pedidos de múltiplas fontes (API e S3) são validados, um evento é disparado, e esse evento aciona um fluxo que processa e armazena o pedido final. Você aprendeu a:

- Criar e configurar regras no Amazon EventBridge para rotear eventos baseados em padrões.
- Usar SQS como um buffer entre o EventBridge e uma Lambda de processamento.
- Desenvolver uma Lambda para consumir de SQS, executar lógica de negócio (simulada) e interagir com DynamoDB para persistência.
- Projetar uma tabela DynamoDB para armazenar dados de pedidos.

Na próxima e última aula, expandiremos a funcionalidade para lidar com outros aspectos do ciclo de vida de um pedido (como cancelamento e alteração), reforçaremos o tratamento de erros com DLQs em todos os pontos e introduziremos o conceito de Lambda Layers para reuso de código.

Prepare-se para a Aula 4!