

Desafio AWS Step Functions

Olá! Eu sou a Bianca, e esse é meu repositório do desafio de **AWS Step Functions**. Aqui eu vou contar um pouco da minha jornada de aprendizado, mostrar meus experimentos e insights, e até aquele momento bagunçado que aconteceu (spoiler: foi aprendizado puro 😅).

Sobre o Desafio

Este projeto tem como objetivo explorar o AWS Step Functions, criando workflows automatizados para orquestrar diferentes serviços da AWS. Durante o aprendizado, testei a integração com serviços como **Lambda**, **EC2**, **DynamoDB** e **EBS**, entendendo como cada serviço pode ser conectado dentro de uma máquina de estado.

Conceitos Explorados

- **Workflow**: a sequência de passos que uma aplicação ou processo deve seguir.
- **State Machine**: a “máquina de estados” que controla cada passo do workflow.
- **Choice**: decisões dentro do fluxo, como um “se isso, faça aquilo”.
- **Buckets**: lugares para armazenar arquivos e dados.
- **Templates**: modelos prontos para agilizar a criação de workflows.

Funcionalidades Exploradas

- Criação de máquinas de estado no AWS Step Functions.
- Integração com **AWS Lambda** para execução de funções serverless.
- Uso de **Amazon DynamoDB** para operações de banco de dados.
- Testes com **EC2** e **EBS** para entender fluxos de computação e armazenamento.
- Exploração de diferentes padrões e APIs do AWS SDK.

Como Funciona

1. O workflow começa no estado **Start**.
2. Executa a função Lambda.
3. Passa por diferentes serviços dependendo do fluxo (DynamoDB, EC2, EBS, etc.).
4. Finaliza no estado **End**.

Nota: O fluxo completo é experimental e foi construído explorando diferentes ações disponíveis no Step Functions, sem um objetivo final específico, mas com foco em aprendizado.

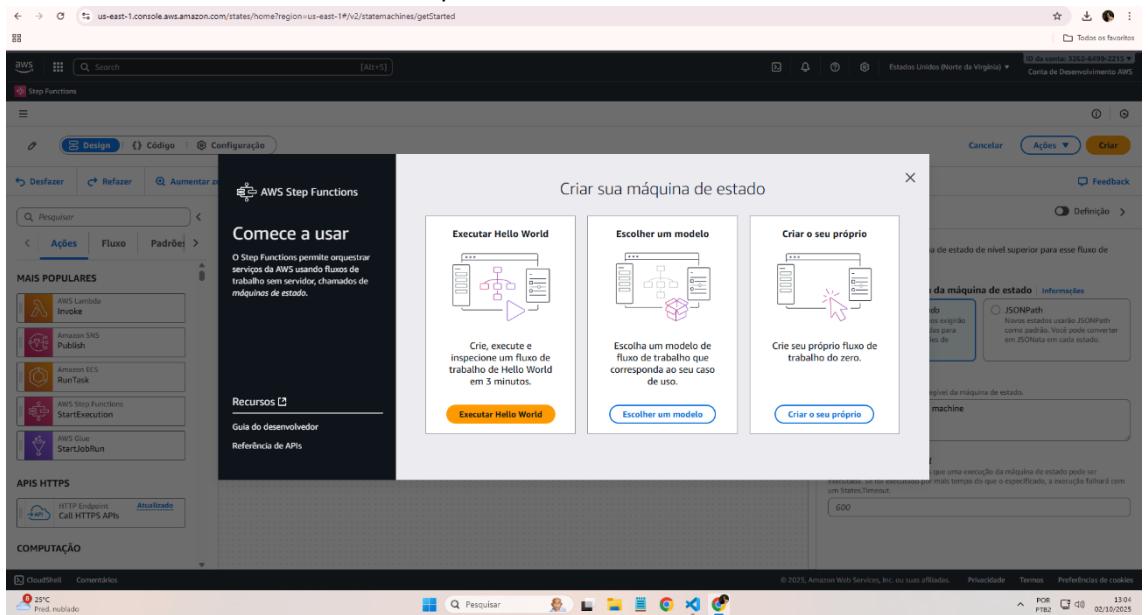
Meu Processo de Aprendizado

- Primeiros testes:** comecei seguindo os passos básicos, entendendo cada elemento do Step Functions.
- Explorando possibilidades:** fui testando e modificando, criando fluxos próprios e tentando entender cada detalhe.
- O “experimento bagunçado”:**

Na terceira imagem, você vai ver que eu estava testando tudo que aparecia na tela, misturando ideias e fluxos. Sim, ficou uma bagunça! Mas foi nesse caos que eu aprendi coisas valiosas sobre como o Step Functions reage a diferentes inputs, erros e decisões. Cada tentativa foi um passo para entender melhor a ferramenta.

Capturas de Tela

- Tela de início do console do AWS Step Functions:



2. Tela do meu modelo de criação:

Fluxo de trabalho

As propriedades da máquina de estado de nível superior trâbalho. [Saiba mais](#)

Linguagem de consulta da máquina de estado

JSONata - recomendado Todos os estados e campos exigirão expressões JSONata válidas para consultas e transformações de dados.

JSONPath Novos es como par em JSON

Comentário - opcional

Uma descrição humanamente legível da máquina de estado.
A description of my state machine

TimeoutSeconds - opcional

O número máximo de segundos que uma execução da máquina executada. Se for executado por mais tempo do que o especificado em States.Timeout.

600

3. Tela da minha experiência montando um workflow:

BatchExecuteStatement

Definição Estado de teste

Configuração **Argumentos e saída**

Linguagem de consulta de estado

Nome do estado BatchExecuteStatement

API DynamoDB: BatchExecuteStatement

Tipo de integração Informações AWS SDK

Observação Você pode passar a entrada para a API Dyr BatchExecuteStatement usando o novo campo Ai na guia Argumentos e Saída.

Aguardar retorno de chamada - opcional

Pause a execução nesse estado até que a execução receba um retorno de chamada de um dos APIs SendTaskSuccess ou SendTaskFailure com o token da tarefa.

4. Tela pra escolher o modelo pronto:

5. Tela do modelo escolhido:

6. Tela do workflow e suas funcionalidades:

Principais Aprendizados

- Aprendi a criar workflows claros e organizados, mesmo quando comecei bagunçando tudo.
- Entendi como usar **Choice States** para tomar decisões automáticas dentro do fluxo.
- Descobri como **debugar erros** e entender porque alguns caminhos não funcionavam.
- Percebi que errar faz parte do aprendizado — e que é divertido quando a bagunça se transforma em conhecimento!

Próximos Passos

- Refinar meus workflows, deixando tudo mais limpo e eficiente.
- Experimentar integrações com outros serviços AWS.
- Continuar explorando e documentando cada insight, para transformar aprendizado em prática real.

Conclusão

Esse desafio foi uma experiência incrível! Mostrou que, mesmo quando tudo parece confuso, a curiosidade e a prática transformam bagunça em conhecimento.