

# Comunicação entre Micro-Serviços AWS Hospedados em LocalStack

- **Serviços AWS Utilizados:**
  - SQS (Simple Queue Service): `aws_sqs_queue`
  - SNS (Simple Notification Service): `aws_sns_topic` e `aws_sns_topic_subscription`
  - Lambda (AWS Lambda): `aws_lambda_function` e `aws_lambda_event_source_mapping`
  - DynamoDB (DynamoDB): `aws_dynamodb_table`
  - API Gateway (API Gateway): `aws_api_gateway_rest_api`, `aws_api_gateway_resource`, `aws_api_gateway_method`, `aws_api_gateway_integration`, `aws_api_gateway_deployment`, `aws_api_gateway_stage`, `aws_lambda_permission`.
  - IAM (Identity and Access Management): `aws_iam_role`, `aws_iam_policy`, `aws_iam_role_policy_attachment`
- **Microserviços:**
  - `ms-communication-bank` (Lambda + API Gateway)
  - `ms-communication-buytrip` (Lambda + API Gateway + SQS trigger)
  - `ms-communication-buyprocess` (Lambda + SQS trigger)
  - `ms-communication-buyfeedback` (Lambda + API Gateway + SQS trigger)
- **Comunicação:**
  - SNS para publicar mensagens em tópicos.
  - SQS para enfileirar mensagens e disparar Lambdas.
  - API Gateway para expor os lambdas `bank`, `buytrip` e `buyfeedback` como API Rest.
- **Armazenamento de Dados:**
  - **DynamoDB** para as tabelas "Cartao" e "Pagamento".
  - **Redis** responsável por armazenar o status das compras das viagens.

## Adaptação do Diagrama Base:

Com base no seu `main.tf`, posso adaptar o diagrama base para refletir sua arquitetura atual. As principais mudanças serão:

- Substituição do MySQL por DynamoDB.
- Substituição do RabbitMQ por SQS e SNS.
- Remoção do Consul.
- Remoção do software de monitoramento de microserviços.
- Adição de ícones AWS para representar os serviços AWS.
- Manutenção do API Gateway.
- Manutenção do Redis, pois ele está na imagem do projeto base, e você informou que ele está no projeto.