

Arquitetura de Integração e Orquestração – Hub de Sistemas

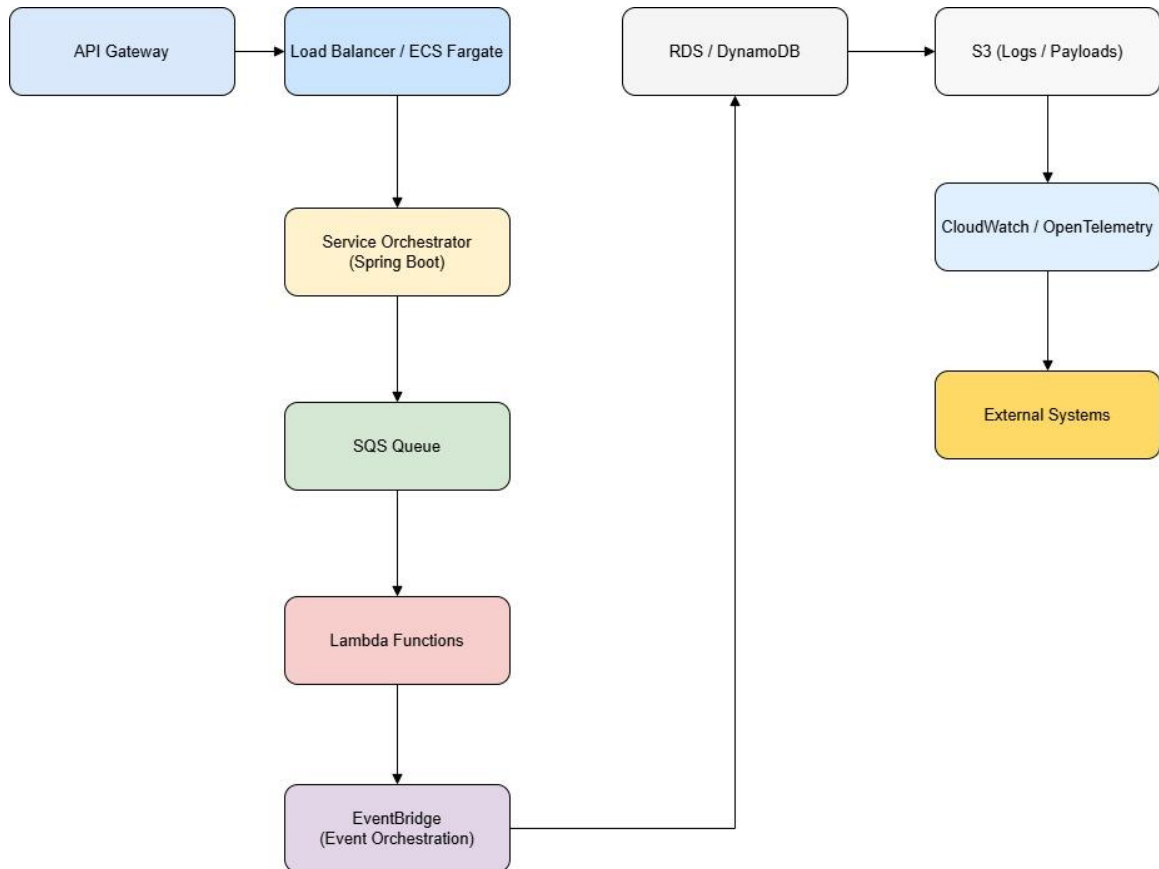
Esta arquitetura foi projetada para atuar como um hub de integração entre múltiplos sistemas, garantindo resiliência, rastreabilidade, escalabilidade e segurança.

O objetivo é permitir que diferentes sistemas se comuniquem de forma eficiente, por meio de APIs e mensagens, com controle de estados, workflows e observabilidade total.

1. Visão Geral da Arquitetura

O diagrama representa o fluxo completo de uma requisição desde a entrada via API Gateway até a comunicação com sistemas externos. Cada componente desempenha um papel específico dentro do ecossistema, com ênfase na orquestração, processamento de mensagens e resiliência.

Diagrama:



2. Passo a Passo do Fluxo de Dados

1 - API Gateway

Responsável por receber as requisições externas (HTTP/REST). Ele atua como ponto de entrada seguro e centralizado, aplicando autenticação, rate limiting e roteamento das chamadas para os serviços internos.

2 - Load Balancer / ECS Fargate

Após o gateway, as requisições são encaminhadas para o balanceador de carga, que distribui o tráfego entre múltiplas instâncias do container rodando no ECS Fargate. Isso garante alta disponibilidade e escala automática conforme a demanda.

3 - Service Orchestrator (Spring Boot)

Este serviço é o núcleo da arquitetura. Implementado em Spring Boot, ele aplica as regras de negócio, valida os dados recebidos e orquestra as etapas seguintes. Também interage com filas e eventos para garantir desacoplamento e tolerância a falhas.

4 - SQS Queue

As mensagens de processamento são enviadas para a fila (SQS), que atua como um buffer entre os serviços. Essa abordagem garante que o sistema continue operando mesmo que algum consumidor temporariamente falhe.

5 - Lambda Functions

Funções serverless que são disparadas por eventos (por exemplo, novas mensagens na fila). Elas executam tarefas específicas e curtas, como transformações de dados, chamadas externas ou notificações.

6 - EventBridge (Event Orchestration)

O EventBridge conecta eventos de diferentes serviços. Ele permite que novos fluxos sejam adicionados sem impactar os sistemas existentes, oferecendo flexibilidade e baixo acoplamento.

7 - Banco de Dados (RDS / DynamoDB)

Os dados transacionais e de workflow são armazenados aqui. O RDS é usado para dados relacionais e o DynamoDB para dados de alta performance ou não estruturados.

8 - S3 (Logs / Payloads)

Usado para armazenar logs, arquivos grandes e payloads recebidos ou enviados para terceiros. Ideal para manter rastreabilidade e auditoria completa.

9 - CloudWatch / OpenTelemetry

Responsáveis pela observabilidade do ambiente. Capturam métricas, logs estruturados e traces de transações ponta a ponta. Permitem monitorar desempenho e detectar falhas rapidamente.

External Systems

São os sistemas externos (terceiros ou legados) que interagem com o hub. A comunicação é feita via APIs, eventos ou integrações específicas. O sistema de orquestração central garante controle e resiliência nessa comunicação.

3. Estratégias e Boas Práticas

- **Alta Disponibilidade:** Uso de ECS Fargate com auto scaling e múltiplas zonas de disponibilidade.
- **Desacoplamento:** Comunicação assíncrona via SQS e EventBridge reduz dependências diretas.
- **Observabilidade:** Integração com CloudWatch e OpenTelemetry para métricas e tracing distribuído.
- **Segurança:** Comunicação HTTPS, autenticação via API Gateway e gerenciamento de segredos via AWS Secrets Manager.
- **Escalabilidade:** Componentes serverless e containers elásticos garantem adaptação à carga.
- **Rastreabilidade:** Logs e payloads armazenados no S3 e correlação de eventos via IDs únicos.

4. Pontos de Atenção e Riscos

- **Latência entre serviços:** Deve ser monitorada com métricas e alertas.
- **Orquestração complexa:** Crescimento excessivo do workflow pode exigir ferramentas adicionais como Step Functions.
- **Custos operacionais:** Funções Lambda e SQS devem ser otimizadas para evitar gastos desnecessários.
- **Gestão de erros:** Estratégias de DLQ (Dead Letter Queue) e retries configuradas no SQS e EventBridge.

5. Conclusão

Essa arquitetura oferece uma base sólida e moderna para integração entre sistemas diversos, unindo resiliência, segurança, escalabilidade e observabilidade.

O uso de componentes AWS desacoplados permite que o sistema evolua de forma incremental, mantendo rastreabilidade completa de ponta a ponta.

Tech Lead:

Wesley Fernando Simonini.