Projeto: Centralização e Análise de Logs do PostgreSQL com Elastic Stack

Versão 1.0 • Escopo: 40+ bancos PostgreSQL • Autor: Equipe de Dados/Observabilidade

1. Sumário Executivo

Este projeto estabelece uma plataforma para coleta, indexação e busca unificada de logs de 40+ bancos PostgreSQL usando Filebeat → Elasticsearch → Kibana. O objetivo é acelerar investigações (erros, deadlocks, slow queries), padronizar retenção (ILM) e permitir observabilidade central sem impactar o banco transacional.

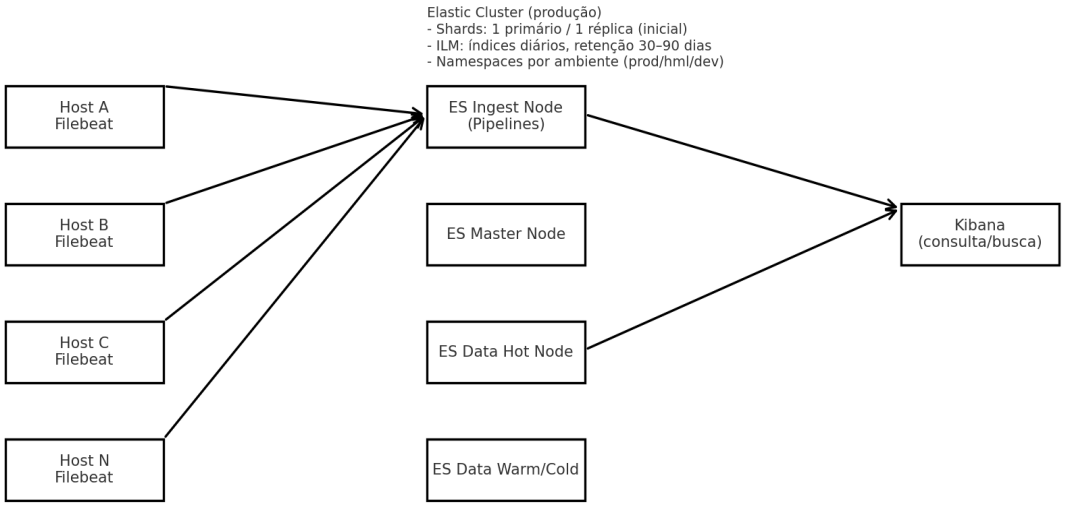
2. Objetivos e Não-Escopo

- Centralizar logs de PostgreSQL em índices diários por ambiente (prod/hml/dev).
- Permitir busca unificada e filtros por banco, cluster, tenant, severidade.
- Implantar políticas de retenção via ILM e boas práticas de shards.
- Disponibilizar dashboards e buscas salvas no Kibana.

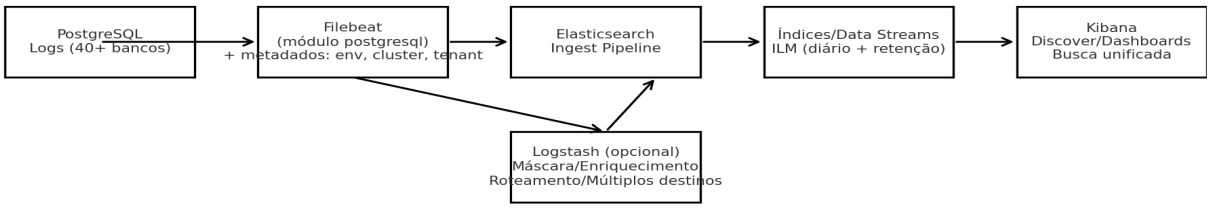
Fora de escopo: métricas de infraestrutura/DB (cobertas por Prometheus+Grafana).

3. Arquitetura da Solução

3.1 Diagrama de Arquitetura (lógico)



3.2 Fluxograma de Integração



3.3 Componentes e Responsabilidades

- **PostgreSQL**: Gera logs conforme postgresql.conf (erros, conexões, duração, autovacuum etc.).
- **Filebeat**: Coleta arquivos de log locais; aplica módulo postgresql; adiciona metadados (env/cluster/tenant); envia ao Elasticsearch.
- **Elasticsearch**: Ingestão, indexação, ILM; data streams/índices diários; busca de texto e agregações.
- **Kibana**: Descoberta, visualização, notebooks e alertas opcionais de log.
- **Logstash (opcional)**: ETL avançado, mascaramento/roteamento/múltiplos destinos (ES + S3/Kafka).

4. Requisitos

Funcionais

- Coletar logs de 40+ instâncias de PostgreSQL.
- Indexar com metadados mínimos: env, db_cluster, tenant, host, database, user, severity.
- Permitir buscas por mensagens, banco, usuário, severidade e janelas de tempo.
- Fornecer retenção configurável (30–90 dias) com exclusão automática.

Não-funcionais

- Alta disponibilidade opcional via réplicas de shards e múltiplos data nodes.
- Segurança: TLS entre agentes e cluster; RBAC no Kibana.
- Observabilidade básica do pipeline (saúde do Filebeat/ingest; crescimento de índices).

5. Estratégia de Indexação e ILM

- Padrão de índices por ambiente: logs-postgresql--yyyy.MM.dd.
- Shards: 1 primário por índice diário (aumentar somente mediante evidência). Réplicas = 1 em produção.
- ILM: rollover diário e retenção de 30–90 dias (ajustável).
- Campos chave para filtro: postgresql.log.database, postgresql.log.user, log.level, fields.env, fields.db_cluster, fields.tenant.

6. Multi-tenant e Organização

- Adicionar campos fields.env, fields.db_cluster, fields.tenant, host.name no Filebeat.
- Data views no Kibana: global e separadas por ambiente.
- Separação por índices distintos por tenant somente quando exigência jurídica/operacional demandar.

7. Segurança

- TLS e autenticação para Filebeat → ES; usuários e roles no Kibana.
- Mascaramento de PII em ingest pipeline ou Logstash (gsub).
- Privilégios mínimos: Filebeat com credencial restrita ao índice de destino.

8. Dimensionamento Inicial (Guidelines)

- Heap do ES ≈ 50% da RAM do processo (máx. ~31 GB).

- Estimativa de armazenamento: volume diário de logs × retenção × fator de compressão (~0.3–0.6).
- Separar nós *ingest* e *data* quando a taxa de ingestão crescer.

9. Etapas de Implementação & Checklists

Fase 1 — Preparação

- Mapear todos os hosts/instâncias de PostgreSQL e caminhos de log.
- Definir env/cluster/tenant por instância.
- Planejar ILM (retenção) e capacidade de disco.

Fase 2 — Infraestrutura

- Subir Elasticsearch (com TLS e usuários) e Kibana.
- Criar templates e políticas ILM.
- Configurar roles/usuários para Filebeat e analistas.

Fase 3 — Agentes (Filebeat)

- Instalar Filebeat em cada host; habilitar módulo postgresql.
- Adicionar fields.env, fields.db_cluster, fields.tenant.
- Direcionar saídas para índices por ambiente.

Fase 4 — Kibana

- Criar data views (global e por ambiente).
- Criar buscas salvas e dashboards básicos.

Fase 5 — Validação

- Verificar ingestão contínua e mapeamento (campos, tipos).
- Testar filtros por banco/usuário/severidade.
- Checar rotação diária e aplicação de ILM.

Fase 6 — Operação

- Revisar crescimento de índices e ajustar retenção.
- Auditar permissões e rotação de credenciais.
- Planejar upgrades e testes de restauração.

10. Estrutura do Repositório (GitHub)

/docs/ – documentação, PDFs, diagramas
/docs/images/ – imagens (fluxograma, arquitetura)
/config/ – templates de índice, ILM, pipelines de ingest (sem segredos)
/infra/ – manifests/compose e instruções de provisionamento
/examples/ – exemplos de buscas KQL, data views, dashboards exportados
README.md – visão geral do projeto
LICENSE – licença

11. Riscos & Mitigações

- Crescimento inesperado de volume de logs → monitorar tamanho diário e ajustar ILM/retention.
- Campos inconsistentes entre fontes → padronizar via módulo postgresql (ECS) e pipelines.
- Dados sensíveis em logs → mascarar com Ingest Pipeline/Logstash; revisão de logging no app/DB.

12. Critérios de Aceite

- Logs de todas as 40+ instâncias visíveis em data view global.
- Busca por banco/usuário/severidade funcionando.
- Índices diários com rotação automática e retenção aplicada.