

NÍVEL AVANÇADO



PROJETO 13

(CONTEÚDO DISPONÍVEL) {
MICROSERVICE;
AGREGADOR;
DE LOGS;
(end);
})();

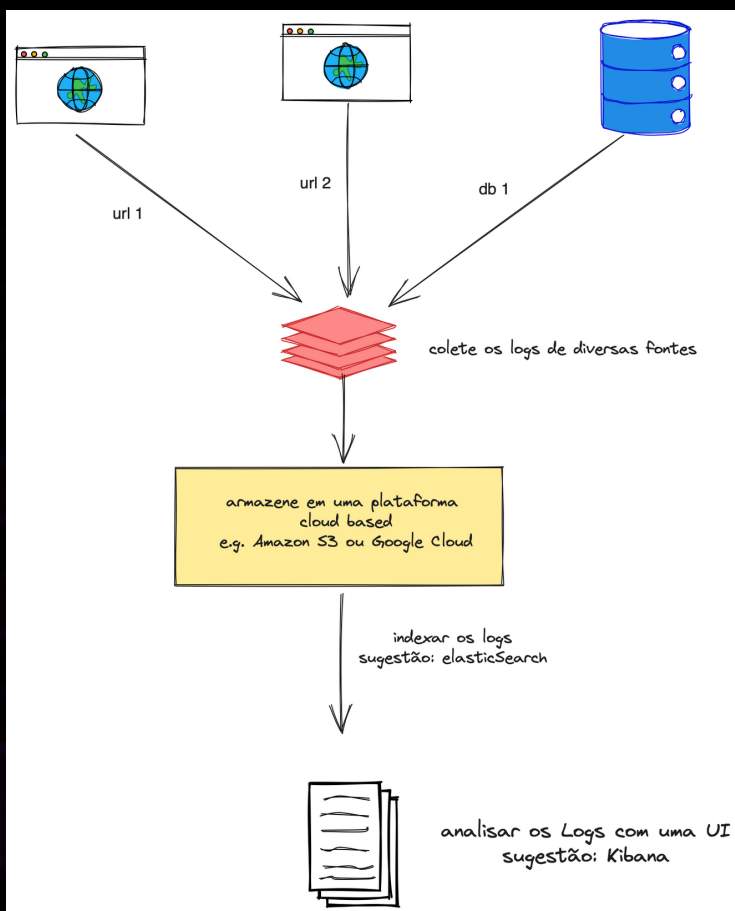
#PORTFÓLIOBOOSTPROGRAM

CONHECIMENTOS REQUIRIDOS:



BACK-END

WIREFRAME



MICROSERVICE AGREGADOR DE LOGS

Cada conexão com seu servidor web é **registrada**; sempre que um usuário solicita um recurso, uma linha no arquivo de log é gravada. Você pode usar esses tipos de **logs** para obter uma visão muito precisa do **tráfego** que chega ao seu site.

No entanto, muitas das vezes nos deparamos com logs em serviços separados e de forma desorganizada

TECH STACK



NodeJS

LIBRARIES



ElasticSearch



LogStash



BRIEFING

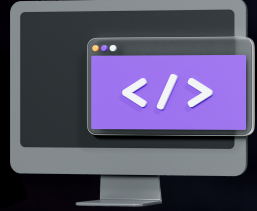
Digamos que você tenha um aplicativo da web que esteja gerando muitos **logs**. Esses logs podem vir de fontes diferentes, como o **servidor da Web**, o **banco de dados** e o **próprio aplicativo**. Se você não **agregar** esses logs, eles ficarão espalhados por todo o lugar, dificultando sua **localização e análise**.

Para agregar logs, você pode usar uma ferramenta de **agregação de log**. Esta ferramenta irá coletar logs de diferentes fontes, armazená-los em um local central e torná-los **pesquisáveis**. Isso facilitará a **localização e a análise de logs** e também o ajudará a **solucionar problemas**.

NÍVEL ÚNICO

Implementar a solução utilizando **nodeJS**.

REQUISITOS DETALHADOS



- ➔ Instale as dependências necessárias. Você precisará **instalar** as seguintes dependências: **npm install logstash**, **npm install elasticsearch**. Crie um arquivo de configuração **logstash**.
- ➔ Esse arquivo informará ao Logstash de onde **coletar logs** e onde **armazená-los**. Crie um **index elasticsearch**. Esse índice armazenará os logs coletados pelo **Logstash**. Inicie o Logstash.
- ➔ Isso iniciará a **coleta de logs** das fontes que você especificou no arquivo de configuração. Inicie o **Elasticsearch**. Isso iniciará a **indexação dos logs** coletados pelo **Logstash**. Acesse os registros
- ➔ Você pode acessar os **logs armazenados** no **Elasticsearch** usando a **API do Elasticsearch**. Segue um exemplo de um arquivo de configuração do **logStash**:

```
input {
  tcp {
    port => 5000
  }
}

output {
  elasticsearch {
    hosts => ["localhost:9200"]
  }
}
```


➔ Este arquivo de configuração informa ao Logstash para escutar na porta 5000 para **conexões TCP**. Quando uma conexão é recebida, o Logstash encaminhará os logs para o **Elasticsearch**.

➔ Um exemplo do elasticSearch index:

```
{ "index": "logs", "mappings": { "_default_": {  
  "properties": { "message": { "type": "text" } } } }
```

➔ Esse índice diz ao **Elasticsearch** para **armazenar** os **logs** em um campo chamado **message**. O campo de mensagem é um campo de **texto**, o que significa que pode armazenar qualquer tipo de texto.

➔ Depois de seguir essas etapas, você terá um sistema de **agregação de log** em funcionamento usando o **Node.js**. Você pode usar este sistema para **coletar logs** de seus aplicativos e **armazená-los** em um local central.

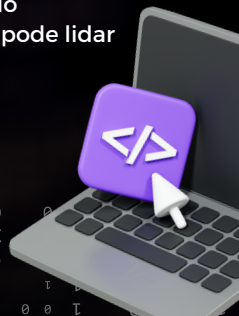
Registros de indexação

➔ Ao indexar logs, você está essencialmente criando um **banco de dados pesquisável** dos **logs**. Isso facilita a localização de logs específicos pesquisando **palavras-chave** ou **padrões**.

➔ Você também pode usar o **índice** para gerar **relatórios**, que podem ajudá-lo a acompanhar o **desempenho** de seu aplicativo ou **solucionar problemas**.

➔ Existem algumas maneiras diferentes de **indexar logs**. Uma maneira comum é usar uma ferramenta como o **Elasticsearch**.

➔ O Elasticsearch é um **mecanismo de pesquisa** projetado especificamente para **indexar logs**. É muito eficiente e pode lidar com um grande volume de logs.



Analizando registros

- ➡ Depois que os logs forem **indexados**, você poderá começar a **analisá-los**. Isso pode envolver a procura de **padrões** nos logs ou a **geração de relatórios**.
- ➡ Por exemplo, você pode procurar **padrões nos logs** que indicam que seu aplicativo está sobrecarregado. Você também pode gerar relatórios que mostram quantos erros seu aplicativo está gerando.
- ➡ Há várias **ferramentas** diferentes que você pode usar para analisar logs. Uma ferramenta popular é o **Kibana**.
- ➡ Kibana é uma ferramenta de **visualização** que pode ser usada para criar painéis interativos de seus logs. **Isso facilita a visualização dos dados em seus logs e a identificação de quaisquer problemas**.

Armazenando logs para retenção de longo prazo

- ➡ Finalmente, você precisa **armazenar os logs** para **retenção de longo prazo**. Isso permitirá que você volte e **analisar os logs**, se necessário.
- ➡ Você pode **armazenar os logs** em um **sistema de arquivos local** ou em um **serviço de armazenamento baseado em nuvem** (amazon ou google).
- ➡ É importante armazenar os logs em um local **seguro**. Você também deve certificar-se de que os logs sejam **copiados regularmente**. Isso ajudará a **proteger seus dados** em caso de desastre.