CONTAINER ELASTIC STACK PARA VISUALIZAÇÃO DE LOGS DO PROXY SQUID

Autor: Thiago Murilo Diniz <thiagodiniz.info at gmail.com>
Data: 28/07/2017

INTRODUÇÃO

Muitas vezes um profissional de TI precisa monitorar, analisar e/ou apresentar dados gerados por uma determinada fonte, como os logs de um serviço ou aplicação. Devem existir excelentes ferramentas, open source ou proprietárias, para auxiliar nesta tarefa, mas que podem ser pouco flexíveis e/ou não atender totalmente à necessidade.

Compartilho neste artigo, a minha primeira experiência com o projeto Open Source *Elastic Stack*, um conjunto de ferramentas para coleta, tratamento e exibição de logs. Demonstrarei como utilizei o Elastic Stack para coletar, tratar e apresentar os logs de acesso do *Squid*.

Breve descrição retirada do site da Elastic:

> Open Source Elastic Stack: Obtenha dados confiáveis e seguros de qualquer fonte, em qualquer formato e procure, analise e visualize em tempo real.

FERRAMENTAS

O Elastic Stack é formado pelas seguintes ferramentas:

`Beats` → são agentes que você instala em seus servidores para enviar diferentes tipos de dados ao Elasticsearch. Os Beats podem enviar dados diretamente ao Elasticsearch ou enviá-los ao Elasticsearch via Logstash, o qual pode ser utilizado para analisar e transformar os dados.

Dentre os Beats disponíveis, utilizaremos o Filebeat, que é um coletor de dados de log para arquivos locais. Instalado como um agente nos servidores, o Filebeat monitora os diretórios de log ou arquivos de log específicos e encaminha o conteúdo ao Elasticsearch (diretamente ou via Logstash) para indexação, atuando como um centralizador de logs. Neste artigo, o Filebeat será responsável por monitorar e encaminhar ao Logstash o conteúdo do arquivo de log de acesso do Squid.

`Logstash` → é um mecanismo de coleta de dados com capacidades de pipeline de tempo real. O Logstash pode unir dinamicamente dados de fontes diferentes e normalizar os dados em destinos de sua escolha. Neste artigo, o Logstash receberá e normalizará os logs do Squid, os enviando posteriormente ao Elasticsearch.

`Elasticsearch` → é um mecanismo de pesquisa e análise distribuído baseado em JSON, projetado para escalabilidade horizontal, máxima confiabilidade e gerenciamento fácil. Permite armazenar, pesquisar e analisar rapidamente grandes volumes de dados. Ele será o responsável por armazenar e indexar os dados enviados pelos Logstash, permitindo buscar estes dados praticamente em tempo real.

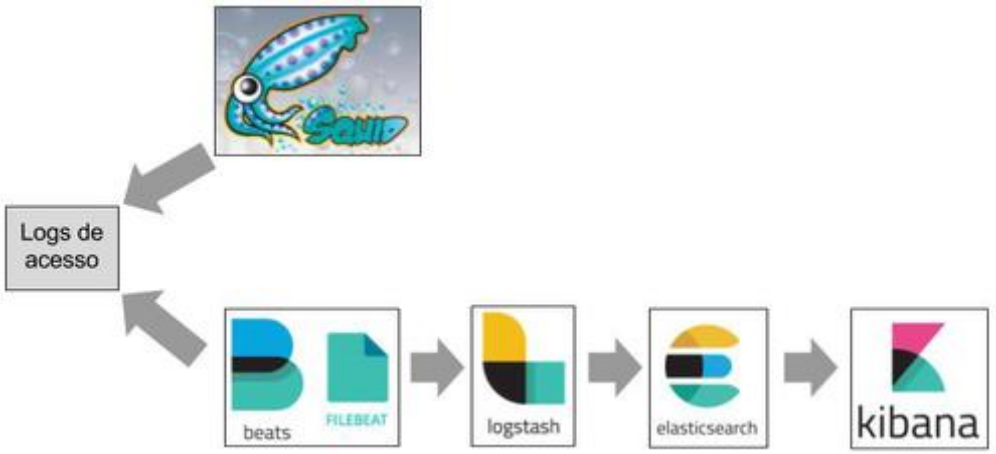
Seguem alguns conceitos importantes referente ao Elasticsearch:

- Index (índice): é como uma tabela de um banco de dados relacional. É uma coleção de documentos com características semelhantes. Por exemplo, podemos ter um índice para dados de clientes, outro para catálogo de produtos e outro para dados de pedidos. Um índice é identificado por um nome (com letras minúsculas) que é usado para se referir ao índice ao executar operações de indexação, pesquisa, atualização e exclusão de documentos.
- Type (tipo): Dentro de um índice você pode definir um ou mais tipos de documentos. Um tipo é uma categoria/partição lógica do seu índice. Em geral, um tipo é definido para documentos que possuem um conjunto de campos comuns. Por exemplo, vamos assumir que você possua uma plataforma de blogs e armazena todos os seus dados em um único índice. Neste índice você pode definir um tipo de dados de usuário, outro de postagens e outro de comentários, cada tipo com seus campos (fields) específicos.
- Document (documento): é uma unidade básica de informação que pode ser indexada. É como uma linha em uma tabela de um banco de dados relacional. Por exemplo, você pode ter um documento para um único cliente e outro para um único produto. Este documento é expresso em JSON. Dentro de um índice/tipo você pode armazenar quantos documentos desejar. Observe que, embora um documento resida fisicamente em um índice, ele é indexado/atribuído a um tipo dentro de um índice. Por exemplo, armazenei um novo "documento" no "tipo comentário" do "índice blog".
- Fields and Datatypes (campos e tipos de dados): um documento possui uma lista de campos. Um campo é como uma coluna em uma tabela de um banco de dados relacional. Cada campo pode armazenar um determinado tipo de dado que pode ser mapeado dinamicamente ou explicitamente.
- Index Templates (modelos de índice): permitem que você defina modelos de configuração e mapeamento que serão automaticamente aplicados quando novos índices forem criados.

Kibana → é uma plataforma de análise e visualização projetada para trabalhar com o Elasticsearch, além de permitir configurar e gerenciar todos os aspectos do Elastic Stack. Você usa o Kibana para pesquisar, visualizar e interagir com dados armazenados em índices Elasticsearch. Você pode facilmente realizar análises avançadas de dados e visualizá-las em uma variedade de gráficos, tabelas e mapas.

Utilizaremos o Kibana para visualizar as informações armazenadas e indexadas no Elasticsearch.

Pode-se observar na Figura 1 o fluxo da comunicação entre os mecanismos do Elastic Stack com os logs de acesso do Squid.



(//img.vivaolinux.com.br/imagens/artigos/comunidade/lab_stack_detail.jpg)

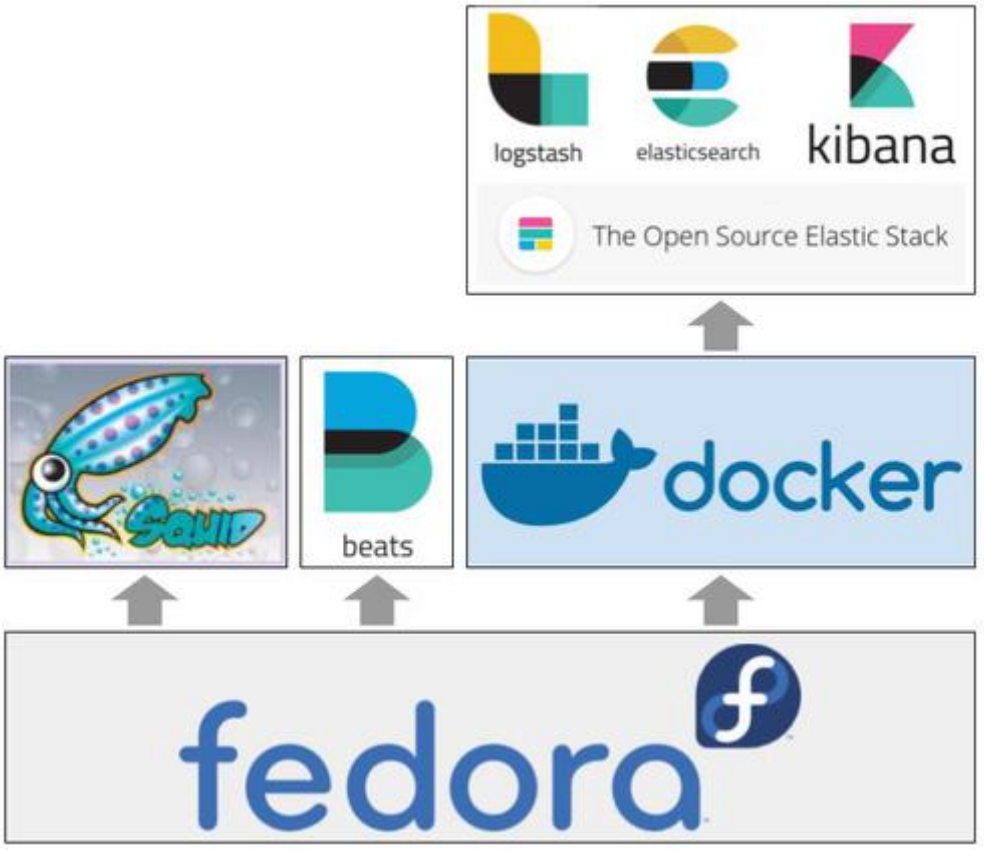
Figura 1

PREPARANDO O AMBIENTE

Para executar e testar o cenário a ser desenvolvido durante o artigo, utilizei apenas meu notebook pessoal, executando a distro Fedora. Você pode seguir a estrutura do cenário utilizado no artigo ou adaptar ao seu cenário, por exemplo, caso você já tenha um servidor *Proxy Squid* em operação e queira executar o *Elastic Stack* em outro servidor na rede.

Na Figura 2 é possível observar a estrutura utilizada:

- instalei o Squid, o Beat Filebeat e o Docker no Fedora;
- instalei o Elastic Stack num container Docker.



(//img.vivaolinux.com.br/imagens/artigos/comunidade/lab_stack.jpg)
Figura 2

Como montei o cenário utilizando Fedora, os comandos usados neste artigo serão baseados nesta distro. Caso você utilize outra distro, adapte os comandos conforme necessidade.

Passos executados para instalar o Squid e aplicar uma configuração básica.

Instalação do Squid:

```
# dnf install squid
```

Configuração do Squid: edite o arquivo */etc/squid/squid.conf* para descomentar a linha:

```
cache_dir ufs /var/spool/squid 100 16 256
```

Crie a estrutura de diretórios do cache:

```
# squid -z
```

Inicie o serviço e verifique seu status:

```
# systemctl start squid.service
```

```
# systemctl status squid.service
```

Passos executados para instalação do Docker, que nos permitirá rodar o Elastic Stack num container.

Instalação do Docker:

```
# dnf install docker
```

Inicie o serviço e verifique seu status:

```
# systemctl start docker
# systemctl status docker
```

SUBINDO O ELASTIC STACK

Agora instalaremos as ferramentas que compõem o *Elastic Stack*.

O *Docker* permitirá subir um container com Logstash, Elasticsearch e Kibana já integrados e prontos para utilizarmos. Mas, caso você queira instalar manualmente o ambiente, basta consultar os manuais disponíveis no site da Elastic:

- Elastic Stack and Product Documentation | Elastic (https://www.elastic.co/guide/index.html)

Utilizaremos a imagem Docker disponibilizada por Sébastien Pujadas (https://hub.docker.com/r/sebp/elk/). A imagem possui uma boa documentação, disponível aqui (http://elk-docker.readthedocs.io).

Vamos baixar a imagem em questão:

```
# docker pull sebp/elk
```

Como requisito do Elasticsearch, para evitar erros de "out of memory" e permitir o armazenamento adequado dos índices, antes de prosseguir verifique na sua distro o valor configurado para o parâmetro "vm.max_map_count":

```
# sysctl vm.max_map_count
```

Se o valor for menor do que 262144, altere com o comando:

```
# sysctl -w vm.max_map_count=262144
```

Caso queira manter esta alteração após o reboot, insira a configuração no arquivo */etc/sysctl.conf*.

Agora, subiremos um container baseado na imagem Docker baixada anteriormente:

```
# docker run -p 5601:5601 -p 9200:9200 -p 5044:5044 -it --name elk sebp/elk
```

A primeira execução do container é mais demorada devido à configuração inicial dos serviços. O shell no qual você executou o container ficará exibindo os logs do Elastic Stack. O container publicará no localhost às portas 5601/tcp para a interface web do Kibana, 9200/tcp para a interface JSON do Elasticsearch e 5044/tcp para o Logstash receber os logs do Filebeat.

Passos executados para instalação do Beat Filebeat. É importante instalar a versão do Filebeat idêntica à versão utilizada da imagem Docker do Elastic Stack, neste caso a 5.4.0 (https://www.elastic.co/downloads/past-releases/filebeat-5-4-0).

Faça download do pacote:

```
$ curl -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-5.4.0-x86_64.rpm
```

Instale o Filebeat:

```
# rpm -ivh filebeat-5.4.0-x86_64.rpm
```

Não inicie o serviço do Filebeat ainda, pois antes temos que configurar o Logstash e o Elasticsearch para receberem adequadamente os logs do Squid.

CONFIGURANDO O ELASTIC STACK

Para configurar o Filebeat, edite o arquivo */etc/filebeat/filebeat.yml* de forma que contenha este conteúdo (https://github.com/thiagomdiniz/vol/blob/master/filebeat.yml).

Com esta configuração, estamos dizendo ao Filebeat para que envie os logs do arquivo */var/log/squid/access.log*, do tipo "squid", para o Logstash que está na porta 5044 do host elk. Como padrão do Filebeat, os índices criados no Elasticsearch terão o nome no formato "filebeat-%{+yyyy.MM.dd}" (por exemplo, "filebeat-2015.04.26").

Dois pontos de atenção antes de prosseguir:

Certificado → a imagem Docker utilizada já vem com um certificado pré-configurado para comunicação segura entre o Filebeat e Logstash (https://github.com/thiagomdiniz/vol/blob/master/logstash-beats.crt). Crie o certificado no local configurado no arquivo */etc/filebeat/filebeat.yml*, no caso */etc/pki/tls/certs/logstash-beats.crt*.

Hostname → ao utilizar este certificado pré-configurado, sua distro deve resolver o nome "elk" para o IP do Logstash, no caso 127.0.0.1 (localhost). Para isso, no arquivo */etc/hosts*, adicione ao fim linha que configura os nomes para o IP 127.0.0.1 o nome "elk". Esta linha na minha distro ficou assim:

```
127.0.0.1   localhost   localhost.localdomain localhost4 localhost4.localdomain4 elk
```

Caso você queira gerar outro certificado e/ou alterar configurações de segurança, pode seguir as orientações na documentação da imagem Docker (http://elk-docker.readthedocs.io/#security-considerations).

Agora, criaremos no Elasticsearch o modelo de índice "filebeat", disponibilizado com o Filebeat (*/etc/filebeat/filebeat.template.json*):

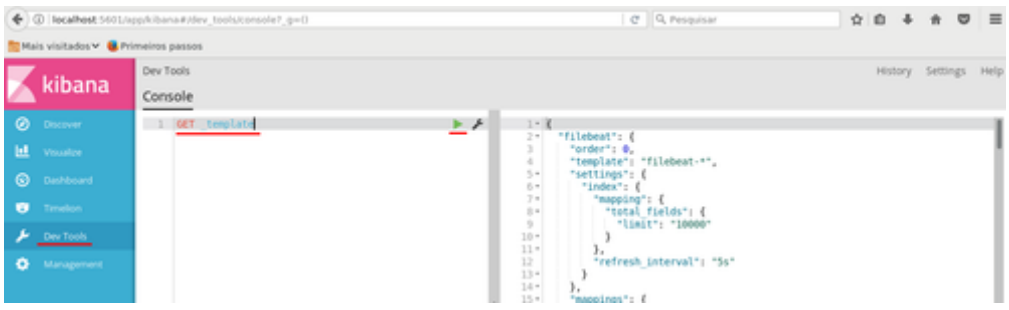
```
$ curl -XPUT 'http://elk:9200/_template/filebeat?pretty' -d@/etc/filebeat/filebeat.template.json
```

Caso o modelo de índice tenha sido criado e enviado com sucesso ao Elasticsearch, você receberá o retorno:

```
{
  "acknowledged" : true
}
```

Você recorda que o Elasticsearch é baseado em JSON e que o *Kibana* permite configurar e gerenciar o Elastic Stack? Então, vamos acessar a interface web do Kibana e conferir se o modelo de índice enviado está realmente lá.

Acesse via navegador o endereço **`http://localhost:5601`** , visualizando assim a interface web do Kibana que está sendo executado no container Docker. Conforme Figura 3, acesse o menu "Dev Tools", digite o comando "GET _template" na caixa à esquerda do console e clique no botão em forma de "play", conforme destacado na figura. Com isso, o Kibana apresentará na caixa à direita do console o retorno em formato JSON da configuração do modelo de índice.



(//img.vivaolinux.com.br/imagens/artigos/comunidade/kibana_dev1.png)

Figura 3 - Consultando modelos de índice no Kibana

Este modelo de índice que acompanha o Filebeat possui alguns types para logs de serviços, como Nginx e Apache. Porém, como não há um type para o Squid, vamos enviar ao Elasticsearch um modelo de índice para configurar o "type" squid e assim armazenar corretamente os "documents" no "index" "filebeat-*".

Para fazer isso, basta colar o conteúdo deste arquivo (https://github.com/thiagomdiniz/vol/blob/master/filebeat_squid) na caixa à esquerda do console do Kibana e clicar no botão em forma de "play" (Figura 4):

(//img.vivaolinux.com.br/imagens/artigos/comunidade/kibana_dev2.png)

Figura 4 - Aplicando modelo de índice via Kibana

Perceba que, conforme parâmetro "template" acima, este modelo será aplicado sempre que for criado um índice de nome que case com o padrão "filebeat-*". Como já sabemos, o Filebeat por padrão criará índices com o prefixo "filebeat-" no nome.

Também definimos a categoria (type) "squid" com os campos "squid.access.geoip.location" para receber dados do tipo "geo_point" e "squid.access.src_ip" e "squid.access.dst_ip" para receber dados do tipo IP. Os demais campos respeitarão a definição do modelo de índice padrão do Filebeat, ou seja, receberão dados do tipo "keyword".

Agora, para configurarmos o Logstash, você precisará acessar o shell do container Docker. Para isso, execute o comando:

```
# docker exec -it elk /bin/bash
```

Acesse o diretório de configuração do Logstash:

```
# cd /etc/logstash/conf.d
```

Crie neste diretório o arquivo de configuração "squid.conf", responsável por dizer ao Logstash como tratar os logs do Squid enviados pelo Filebeat. O conteúdo do arquivo deve ser este (https://github.com/thiagomdiniz/vol/blob/master/squid.conf).

Entendendo o que esta configuração diz ao Logstash: criamos um filtro no qual, caso a mensagem recebida for do tipo "squid" (atributo "document_type" da configuração do Filebeat), utilizamos o plugin "grok" para estruturar o texto das linhas de log do Squid e depois utilizamos o plugin "geoip" para

armazenar informações geográficas dos endereços IP acessados pelos clientes do Proxy Squid.

Destaque aqui para o plugin "grok", que é atualmente a melhor forma de o Logstash transformar dados desestruturados (texto) em algo estruturado e rastreável. Este plugin é perfeito para logs como Syslog, Apache, Nginx, MySQL e, em geral, qualquer formato de log geralmente escrito para humanos. Você basicamente utiliza expressões regulares para detectar os dados do texto e estruturá-los.

O Logstash possui cerca de 120 padrões prontos (patterns) (https://github.com/logstash-plugins/logstash-patterns-core/tree/master/patterns). Se você precisar de ajuda na construção dos seus próprios patterns, os sites grokdebug.herokuapp (http://grokdebug.herokuapp.com) e grokconstructor.appspot (http://grokconstructor.appspot.com/) serão bastante úteis.

Caso queira verificar os detalhes do "grok", veja a documentação oficial (https://www.elastic.co/guide/en/logstash/current/plugins-filters-grok.html).

Para aplicar a nova configuração do Logstash, reinicie o serviço (ainda no shell do container):

```
# service logstash restart
```

Com a configuração ok, agora podemos iniciar o serviço do Filebeat:

```
# systemctl start filebeat.service
```

Não confunda os shells, pois o Filebeat está instalado no Fedora junto com o Squid, e não no container Docker.

Para que o Squid comece a gerar logs de acesso, você precisa navegar utilizando o proxy. Você pode alterar a configuração de proxy do Firefox (Figura 5) para navegar através do seu Squid:

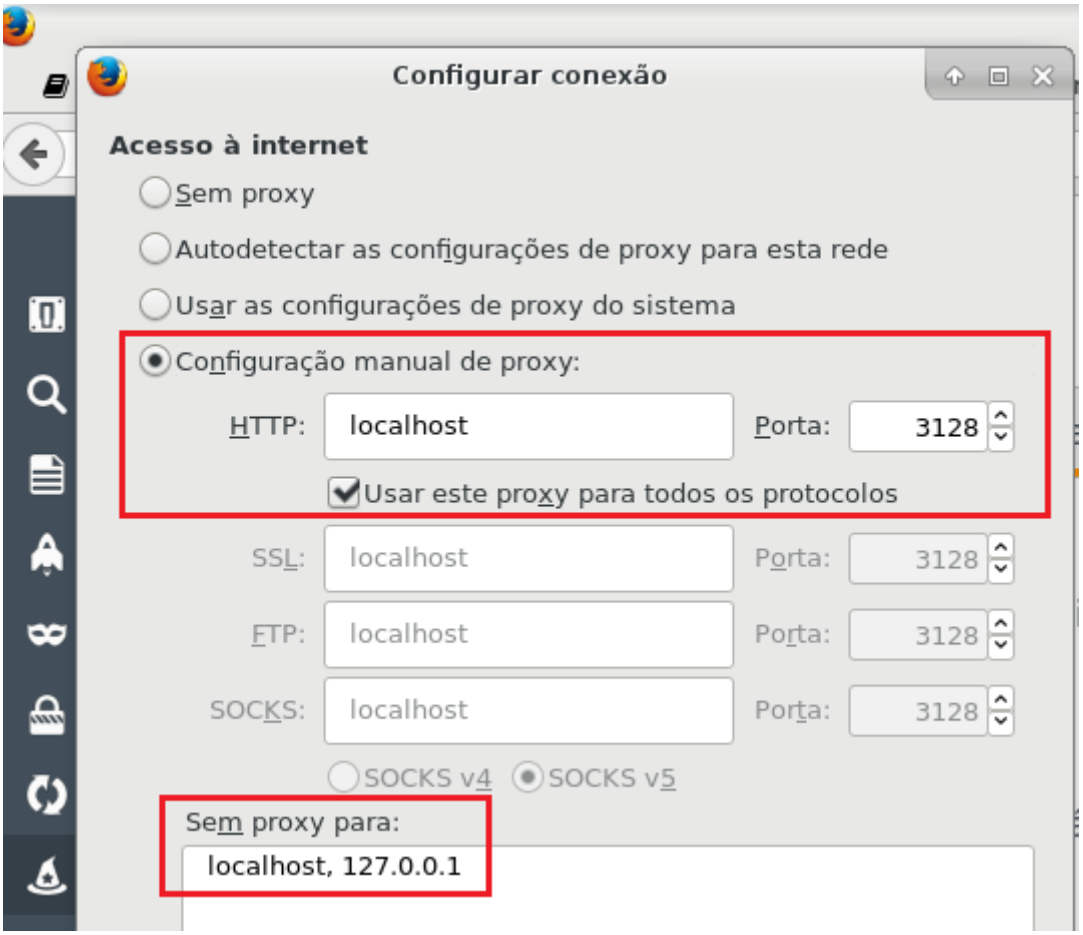


Figura 5 - Configuração de proxy do navegador

A partir do momento em que o Squid começar a gerar logs de acesso, o Filebeat os enviará ao Logstash que por sua vez os normalizará e os enviará ao Elasticsearch. O Elasticsearch criará um novo índice ao receber as informações, com base no template configurado anteriormente.

No momento da criação do índice, você poderá visualizar no log que será exibido no shell do container (Figura 6):

(//img.vivaolinux.com.br/imagens/artigos/comunidade/elastic_index.png)

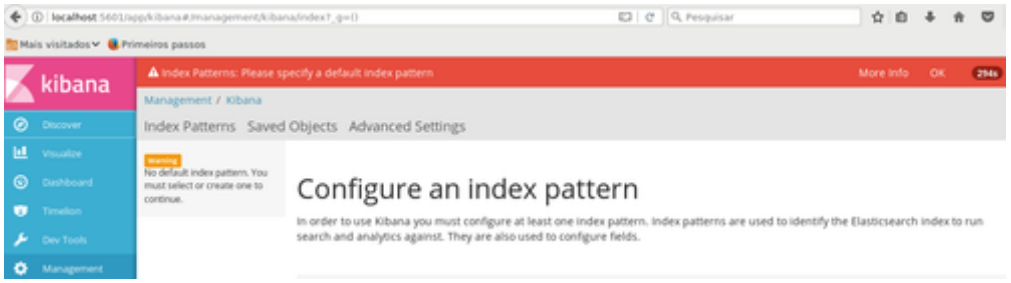
Figura 6 - Índice criado pelo Elasticsearch

VISUALIZANDO OS LOGS NO KIBANA

Agora visualizaremos, através do *Kibana*, os logs do Squid indexados no Elasticsearch.

Acesse a interface web do Kibana (http://localhost:5601) e clique no menu "Discover". Conforme Figura 7, o Kibana o avisará de que ainda não foi definido um "index pattern". O Kibana precisa que, ao menos, um "index pattern" seja configurado para que possa identificar em qual índice do Elasticsearch serão

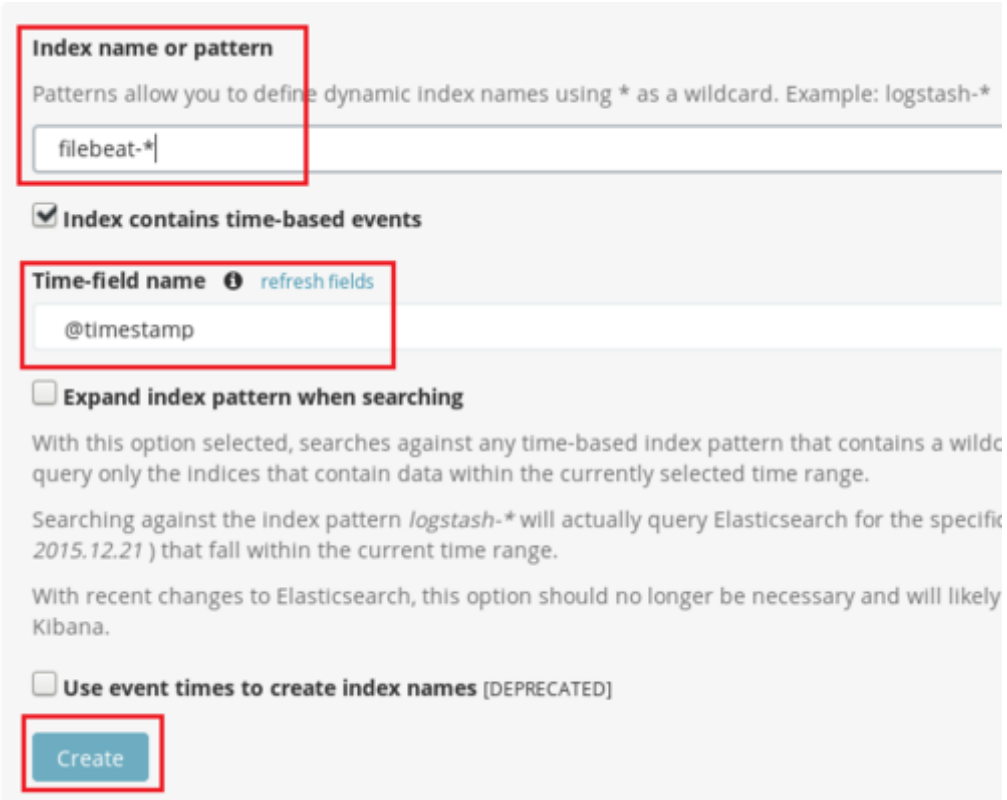
executadas as consultas e análises.



(//img.vivaolinux.com.br/imagens/artigos/comunidade/kibana_warn_pattern.png)

Figura 7 - Kibana sem "index pattern"

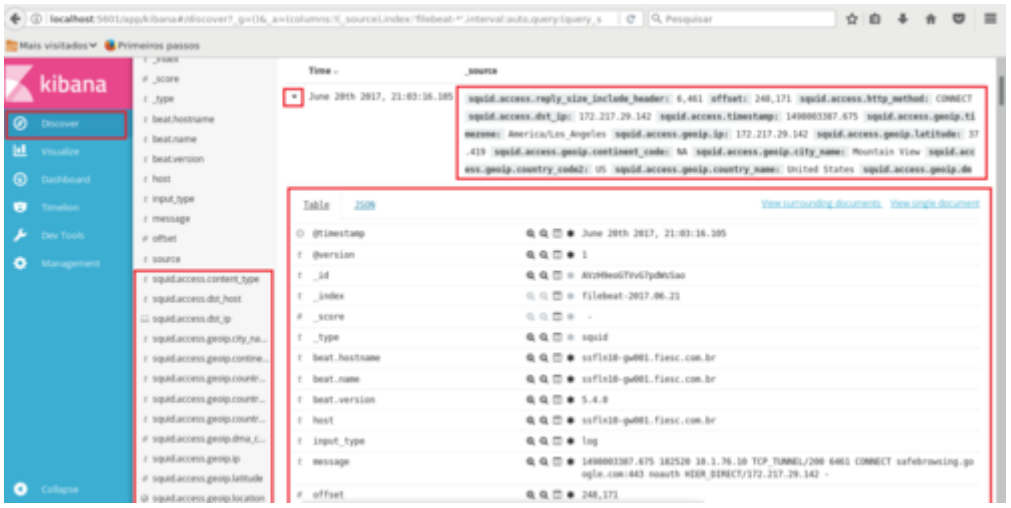
Neste caso, como sabemos que os índices que serão criados no Elasticsearch terão o prefixo "filebeat-", digitaremos filebeat-* no campo "Index name or pattern". Após digitar, aguarde cerca de 3 segundos para que o Kibana atualize o campo "Time-field name" com o valor "@timestamp" e então clique no botão "Create" (Figura 8):



(//img.vivaolinux.com.br/imagens/artigos/comunidade/kibana_pattern.png)

Figura 8 - Configuração de "index pattern" no Kibana

Com isso, clicando novamente no menu "Discover", você visualizará os "documents" indexados no Elasticsearch. Perceba que cada "document" possui os campos mapeados pelo Logstash através do plugin "grok" e também os campos de geolocalização criados pelo plugin "geoip" (Figura 9).

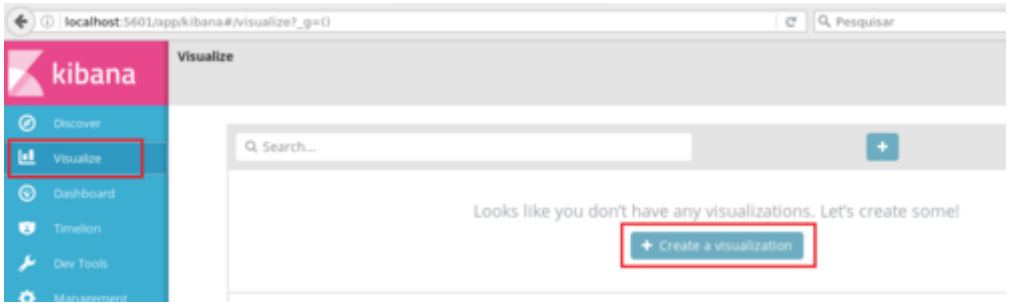


(//img.vivaolinux.com.br/imagens/artigos/comunidade/kibana_discover_document.png)

Figura 9 - Exibição de documento no Kibana

A partir daí, o Kibana nos permitirá gerar visualizações e dashboards para exibição das informações coletadas dos logs de acesso do Squid.

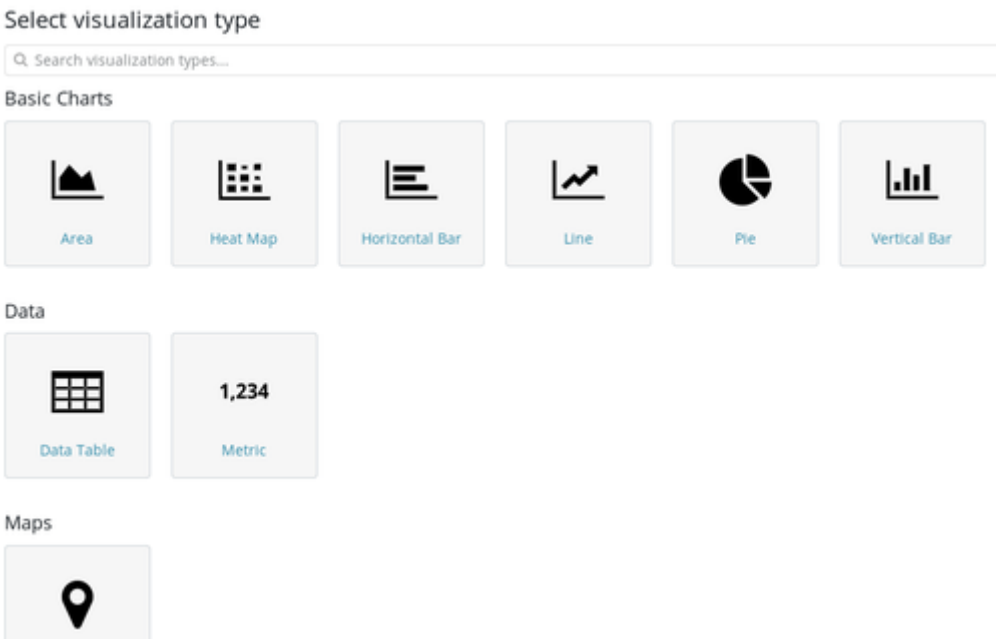
Pode-se observar na Figura 10 o menu "Visualize", que permitirá criar visões como gráficos, tabelas, mapas etc:



(//img.vivaolinux.com.br/imagens/artigos/comunidade/kibana_visualize.png)

Figura 10 - Acessando o menu "Visualize" no Kibana

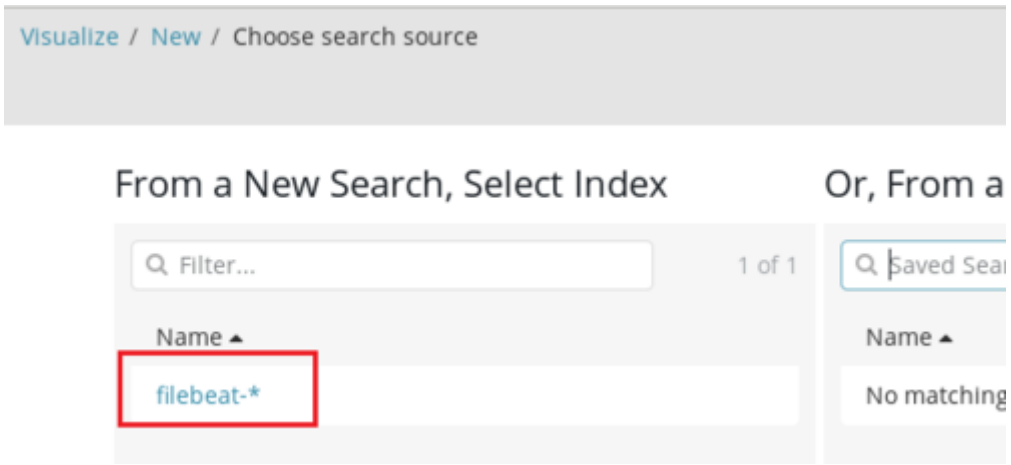
Ao clicar em "Create a visualization" você deverá escolher o tipo de visão que irá criar (Figura 11):



(//img.vivaolinux.com.br/imagens/artigos/comunidade/kibana_vi_type.png)

Figura 11 - Seleção de tipo de visão no Kibana

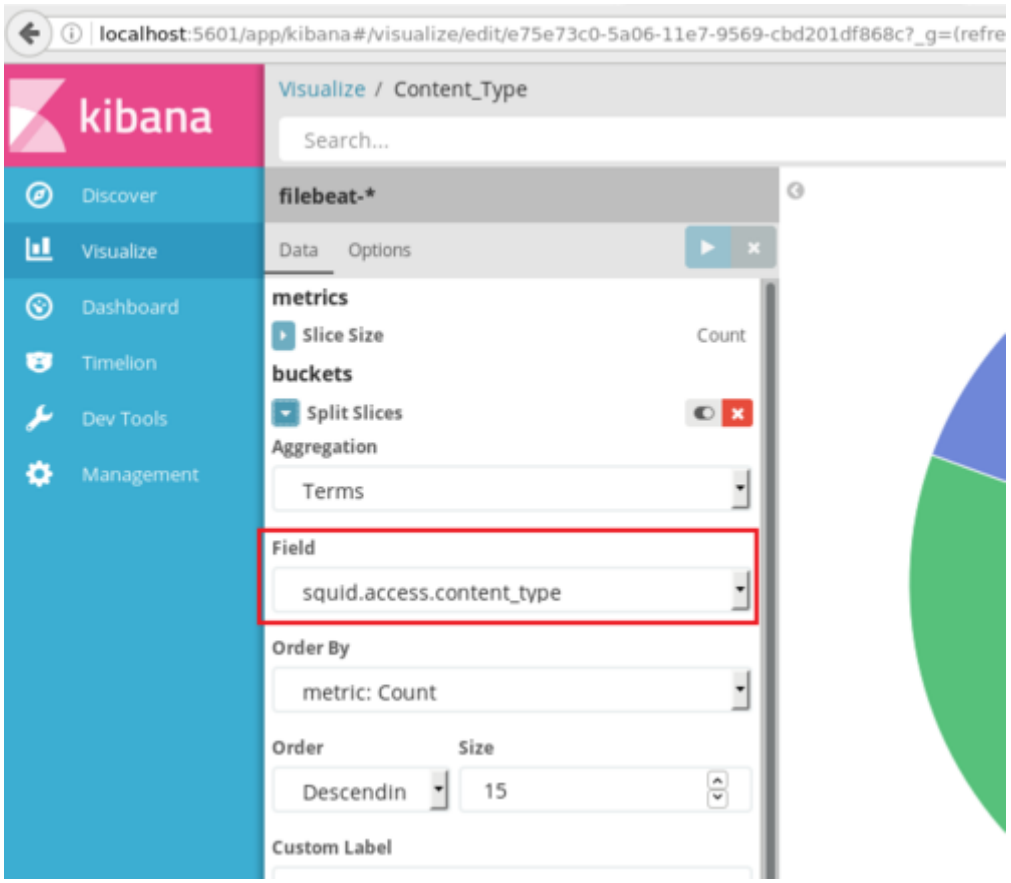
Após selecionar o tipo de visão, selecione o índice do qual as informações serão extraídas (Figura 12):



(//img.vivaolinux.com.br/imagens/artigos/comunidade/kibana_vi_source.png)

Figura 12 - Seleção de origem dos dados no Kibana

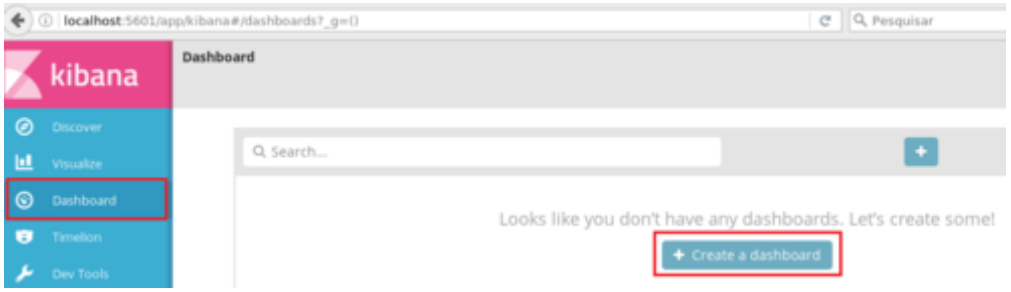
Pode-se observar na Figura 13, que os campos extraídos dos logs de acesso do Squid e indexados no Elasticsearch estão disponíveis para utilização nas visões:



(//img.vivaolinux.com.br/imagens/artigos/comunidade/kibana_vi_fields.png)

Figura 13 - Campos do Squid disponíveis no Kibana

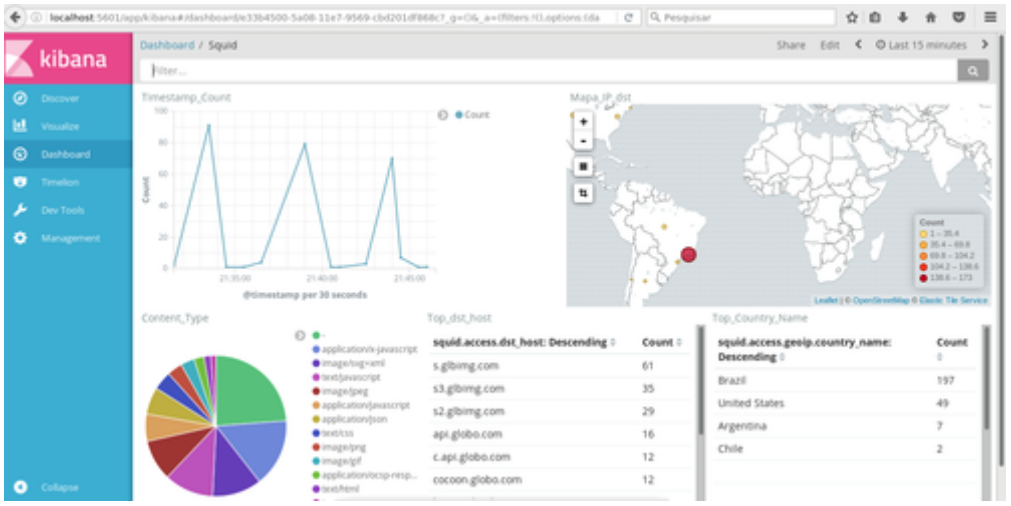
Após criar as visões, você poderá juntá-las em uma só tela criando um dashboard. Para criar um dashboard, clique no menu "Dashboard" e depois em "Create a dashboard" (Figura 14):



(//img.vivaolinux.com.br/imagens/artigos/comunidade/kibana_dashboard.png)

Figura 14 - Acessando o menu "Dashboard" do Kibana

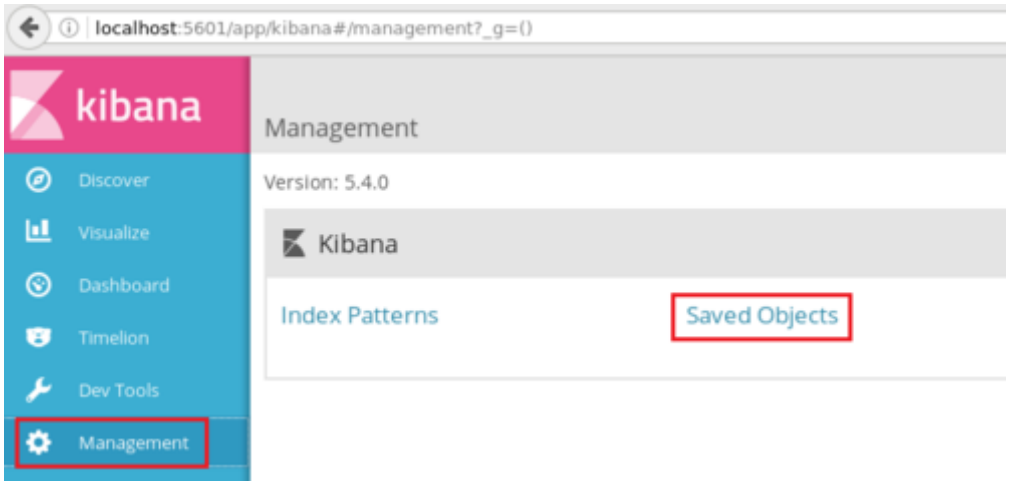
E por fim, você terá um dashboard exibindo as visões adicionadas ao mesmo (Figura 15):



(//img.vivaolinux.com.br/imagens/artigos/comunidade/kibana_dashboard_done.png)

Figura 15 - Dashboard no Kibana

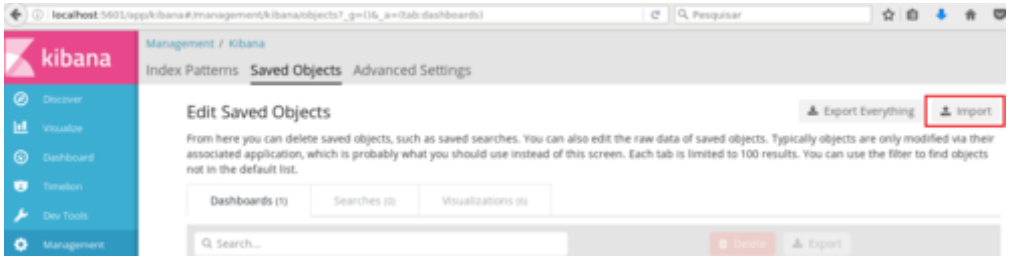
Caso você queira utilizar o dashboard de exemplo que criei na Figura 15, faça download dos arquivos de configuração das visões (https://github.com/thiagomdiniz/vol/blob/master/visualizations.json) e do dashboard (https://github.com/thiagomdiniz/vol/blob/master/dashboard.json), clique no menu "Management" e depois em "Saved Objects" (Figura 16):



(//img.vivaolinux.com.br/imagens/artigos/comunidade/kibana_saved.png)

Figura 16 - Acessando o menu "Management" do Kibana

Depois, conforme Figura 17, clique no botão "Import" e selecione primeiro o arquivo "visualizations.json". Depois clique em "Import" novamente e selecione o arquivo "dashboard.json". Com isso, o dashboard exibido na Figura 15 ficará disponível no menu "Dashboard".



(//img.vivaolinux.com.br/imagens/artigos/comunidade/kibana_import.png)

Figura 17 - Tela que permite importar configurações no Kibana

CONCLUSÃO

Após esta experiência, percebi o quão poderoso é o Elastic Stack e a infinidade de coisas que se pode fazer com estas ferramentas.

Para dar alguns exemplos, poderíamos indexar dados de NetFlow para análise de tráfego de rede, utilizar outros Beats como o PacketBeat ou MetricBeat, gerar alertas com o Watcher ou ainda integrar o Logstash com Nagios ou Zabbix.

REFERÊNCIAS

- Elastic Stack and Product Documentation | Elastic (https://www.elastic.co/guide/index.html)
- https://hub.docker.com/r/sebp/elk/ (https://hub.docker.com/r/sebp/elk/)
- elk-docker (http://elk-docker.readthedocs.io)
- Elasticsearch Architectural Overview - Building VTS (https://buildingvts.com/elasticsearch-architectural-overview-a35d3910e515)
- Anatomy of an Elasticsearch Cluster: Part I - Insight Data (https://blog.insightdatascience.com/anatomy-of-an-elasticsearch-cluster-part-i-7ac9a13b05db)
- Mapping | Elasticsearch Reference [5.5] | Elastic (https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping.html)

Voltar (verArtigo.php?codigo=16500)