Daniel Berman

Table of Contents

oads for its various components since first being introduced, the **ELK** ost popular log management platform. In contrast, Splunk — the space — self-reports 15,000 customers total.

, and why is the software stack seeing such widespread interest and deeper dive.

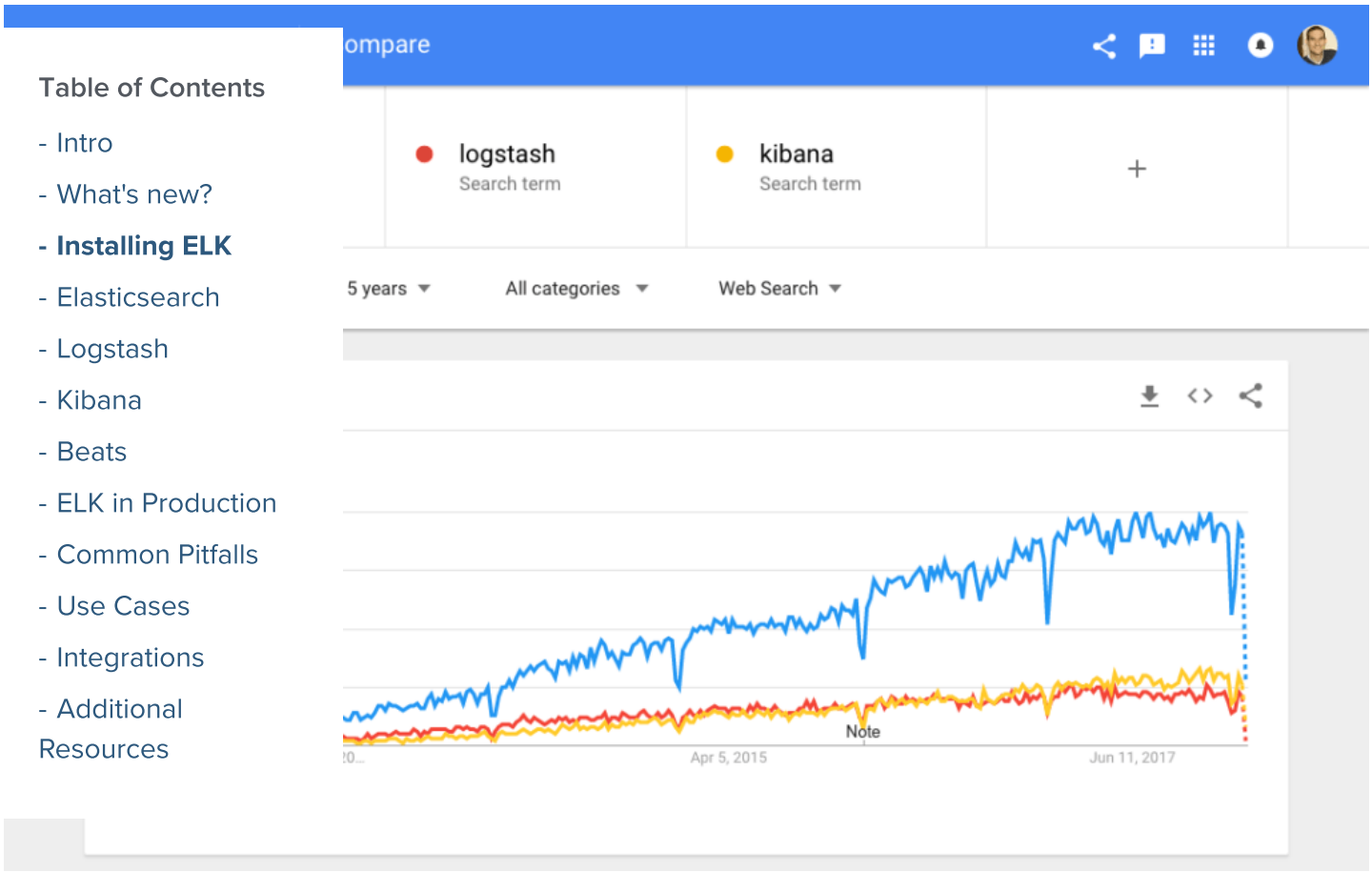
Intro to the ELK Stack

What is the ELK Stack?

The ELK Stack is a collection of three open-source products — Elasticsearch,Logstash, and Kibana — all developed, managed and maintained by Elastic. Elasticsearch is a NoSQL database that is based on the Lucene search engine. Logstash is a log pipeline tool that accepts inputs from various sources, executes different transformations, and exports the

then presents the data in visualizations that provide actionable insights into one's environment.

Why is ELK So Popular?



The ELK Stack is popular because it fulfills a need in the log analytics space. Splunk's enterprise software has long been the market leader, but its numerous functionalities are increasingly not worth the expensive price — especially for smaller companies such as SaaS businesses and tech startups.

general, and this is why newer proprietary log analysis software platforms such as Sumo Logic, which self-reports only 1000 customers, might have a hard time gaining traction today.

After all, how do Netflix, Facebook, Microsoft, LinkedIn, and Cisco monitor their logs? With ELK.

Table of Contents

ysis Becoming More Important?

nfrastructures move to public clouds such as Amazon Web Services oogle Cloud, public cloud security tools, and logging platforms are nd more critical.

uctures, performance isolation is extremely difficult to reach — systems are heavily loaded. The performance of virtual machines in uctuate based on the specific loads, infrastructure servers, number of active users. As a result, reliability and node failures can blems.

orms can monitor all of these infrastructure issues as well as process NGINX and IIS server logs for technical SEO and web traffic analysis, application logs, and ELB and S3 logs on AWS.

In all of these contexts, DevOps engineers, system administrators, site reliability engineers, and developers can all use logs to make better decisions that are data-informed (and not, as Facebook's Adam Mosseri says, data-driven). After all, "big data analytics" is increasingly important for a number of reasons — particularly when it comes to the cloud.

environment”

- Cheaper computational power is allowing engineers to create machine-learning algorithms that can perform predictive analytics in the cloud

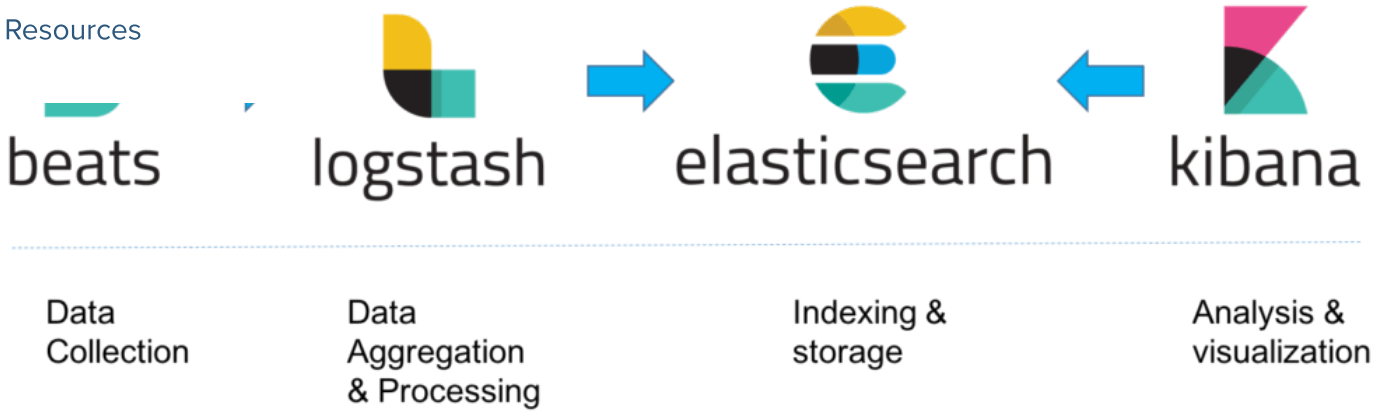
How to Use the ELK Stack for Log Analysis

Table of Contents

the ELK Stack is most commonly used for centralized logging. ntation and maintenance of a production-grade ELK Stack require a nd many additional products. More information on installing and vided in another section of this guide.

ecture

its in the ELK Stack were designed to interact and play nicely with much extra configuration. However, how you end up constructing nds on the environment and specific use case in question. For a ent environment, the classic architecture will look as follows:



ffering | Data Aggregation & Processing | Indexing & storage | Analysis & visualization

Table of Contents

in production can be found in this article.

What's new?

with tips for installation, we'll review the latest Elastic Stack features.

nity are doing a great job of continuously and frequently introducing new capabilities to the stack. The Elasticsearch, Logstash, Kibana and Beats projects especially, are developing rapidly and it's sometimes difficult to keep track with these changes.

In this section, we'd like to highlight the main features introduced in major releases so our readers can stay up-to-date.

experimental in version 6.3). Replacing the old Ruby execution engine, it promises better performance, reduced memory usage and overall — an entirely faster experience.

Kibana

Kibana is undergoing some major facelifting with new pages and usability improvements. The main additions are a new page for infrastructure monitoring, a new page for -time, Canvas, Kibana spaces and a UI for managing Elasticsearch

ıctionbeat (a serverless beat for AWS Lambda) and Journalbeat (for tbeart (for periodic pings on the status of services) went GA.

eakdown of the new features, see this blog post.

Installing ELK

The ELK Stack can be installed using a variety of methods and on a wide array of different operating systems and environments. ELK can be installed locally, on the cloud, using Docker and configuration management systems like Ansible, Puppet, and Chef. The stack can be installed using a tarball or .zip packages or from repositories.

To perform the steps below, we set up a single AWS Ubuntu 16.04 machine on an m4.large instance using its local storage. We started an EC2 instance in the public subnet of a VPC, and then we set up the security group (firewall) to enable access from anywhere using SSH and TCP 5601 (Kibana). Finally, we added a new elastic IP address and associated it with our running instance in order to connect to the internet.

Table of Contents

ersion we installed here is 6.2. Changes have been made in more licensing model, including the inclusion of basic X-Pack features into packages.

lation of Java 8 and higher. The first thing to do is check what Java g.

te your system:

Install java with:

```
1 sudo apt-get install default-jre
```

Checking your Java version now should give you the following output or similar:

```
sudo apt-get install apt-transport-https
```

The next step is to add the repository definition to your system:

```
echo "deb https://artifacts.elastic.co/packages/6.x/apt stable main" | sudo tee -a /etc/a
```


update your repositories and install Elasticsearch:

```
elasticsearch
```

ations are done using a configuration file that allows you to configure
ode name), as well as network settings (e.g. host and port), where
, log files, and more.

we are installing Elasticsearch on AWS, it is a good best practice to
ither a private IP or localhost:

```
csearch/elasticsearch.yml
host"
```

se:

```
sudo service elasticsearch start
```

To confirm that everything is working as expected, point curl or your browser to http://localhost:9200, and you should see something like the following output:

```
13 "build_hash" : "5b1fea5",
14
15 "build_date" : "2018-01-10T02:35:59.208Z",
16
17 "build_snapshot" : false,
18
19 "lucene_version" : "7.1.0",
20
21 "minimum_wire_compatibility_version" : "5.6.0",
```

```
patibility_version" : "5.0.0"
```

```
low, for Search"
```


rch cluster requires a different type of setup. Read our Elasticsearch
e information on that.

sh

ed the repository in the system, all we have to do to install Logstash

```
. logstash
```

Before you run Logstash, you will need to configure a data pipeline. We will get back to that once we've installed and started Kibana.

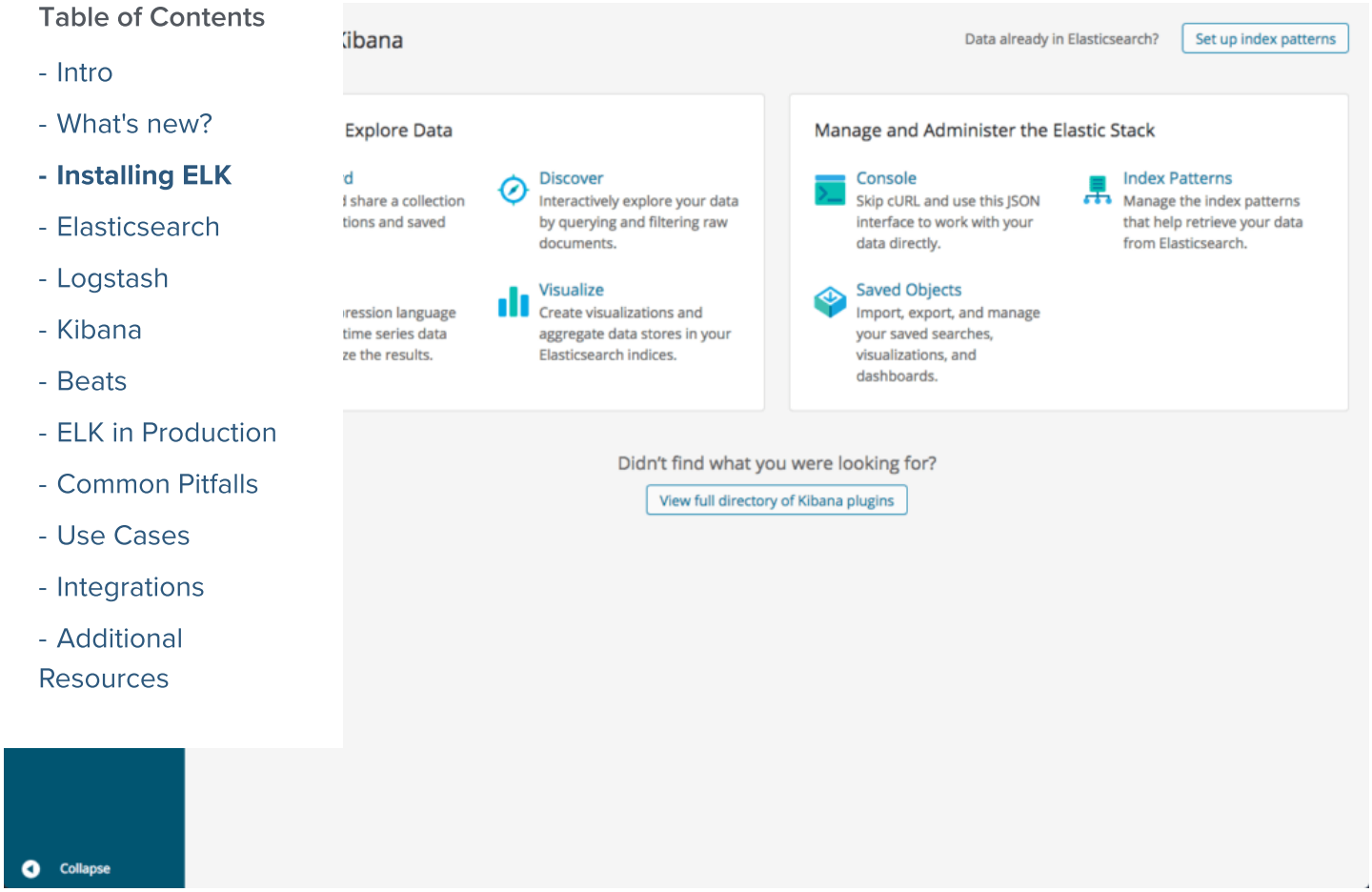
Installing Kibana

As before, we will use a simple apt command to install Kibana:

Now, start Kibana with:

```
sudo service kibana start
```

Open up Kibana in your browser with: http://localhost:5601. You will be presented with the



Installing Beats

To start Metricbeat, enter:

```
1 sudo service metricbeat start
```

Metricbeat will begin monitoring your server and create an Elasticsearch index which you can define in Kibana. In the next step, however, we will describe how to set up a data pipeline using Logstash.

sing the different beats is available on our blog: Filebeat, Metricbeat,

ata

tutorial, we've prepared some sample data containing Apache eshed daily. You can download the data here: https://logz.io/sample-

gstash configuration file at: /etc/logstash/conf.d/apache-01.conf:

```
sh/conf.d/apache-01.conf
```

gstash configuration (change the path to the file you downloaded accordingly):

```
13 }
14
15 filter {
16
17 grok {
18
19 match => { "message" => "%{COMBINEDAPACHELOG}" }
20
21 }
```

Table of Contents


```
mp" , "dd/MMM/yyyy:HH:mm:ss Z" ]

,"

t:9200"]
```

```
1 sudo service logstash start
```

If all goes well, a new Logstash index will be created in Elasticsearch, the pattern of which can now be defined in Kibana.

Table of Contents

Table of Contents

Table of Contents

Congratulations! You have set up your first ELK data pipeline using Elasticsearch, Logstash, and Kibana.

- Installing ELK on Windows
- Installing ELK with Docker
- Installing ELK on Mac OS X
- Installing ELK with Ansible
- Installing ELK on RaspberryPi

Table of Contents

ction of this guide to understand more advanced topics related to arch, Logstash, Kibana and Beats.

Elasticsearch

arch?

ng heart of what is today's the most popular log analytics platform — earch, Logstash, and Kibana). The role played by Elasticsearch is so me synonymous with the name of the stack itself. Used primarily for search and log analysis, Elasticsearch is today one of the most popular database systems available today.

Initially released in 2010, Elasticsearch is a modern search and analytics engine which is based on Apache Lucene. Completely open source and built with Java, Elasticsearch is categorized as a NoSQL database. Elasticsearch stores data in an unstructured way, and

In the context of data analysis, Elasticsearch is used together with the other components in the ELK Stack, Logstash and Kibana, and plays the role of data indexing and storage.

Read more about installing and using Elasticsearch in our Elasticsearch tutorial.

Table of Contents

ch Concepts

re-rich and complex system. Detailing and drilling down into each of possible. However, there are some basic concepts and terms that s should learn and become familiar with. Below are the five “must- t with.

objects that are stored within an Elasticsearch index and are nit of storage. In the world of relational databases, documents can in a table.

me that you are running an e-commerce application. You could have duct or one document per order. There is no limit to how many re in a particular index.

Data in documents is defined with fields comprised of keys and values. A key is the name of the field, and a value can be an item of many different types such as a string, a number, a boolean expression, another object, or an array of values.

Documents also contain reserved fields that constitute the document metadata such as _index, _type and _id

Starting with Elasticsearch 6, an index can have only one mapping type, and this feature will gradually be removed in future versions.

Mapping

Like a schema in the world of relational databases, mapping defines the different types that reside within an index. It defines the fields for documents of a specific type — the data d integer) and how the fields should be indexed and stored in

ned explicitly or generated automatically when a document is es. (Templates include settings and mappings that can be applied index.)

re logical partitions of documents and can be compared to a of relational databases.

nerce app example, you could have one index containing all of the ducts and another with all of the data related to the customers.

indices defined in Elasticsearch as you want but this can affect performance. These, in turn, will hold documents that are unique to each index.

Indices are identified by lowercase names that are used when performing various actions (such as searching and deleting) against the documents that are inside each index.

Shards

performance.

For more information on these terms and additional Elasticsearch concepts, read the 10 Elasticsearch Concepts You Need To Learn article.

eries

Table of Contents

ı top of Apache Lucene and exposes Lucene's query syntax. Getting ntax and its various operators will go a long way in helping you

languages, Elasticsearch supports the AND, OR, and NOT

return events that contain both jack and jill

Will return events that contain ahab but not moby

return events that contain tom or jerry, or both

You might be looking for events where a specific field contains certain terms. You specify that as follows:

- **name:"Ned Stark"**

Ranges

Wildcards, Regexes and Fuzzy Searching

A search would not be a search without the wildcards. You can use the * character for multiple character wildcards or the ? character for single character wildcards.

URI Search

ırch your Elasticsearch cluster is through URI search. You can pass a search using the q query parameter.

ɔ of the iceberg. More information and examples can be found in ries: A Thorough Guide.

ST API

s about Elasticsearch is its extensive REST API which allows you to query the indexed data in countless different ways. Examples of ate with Elasticsearch data are abundant, spanning different ses.

Table of Contents

As extensive as Elasticsearch REST APIs are, there is a learning curve. To get started, read the API conventions, learn about the different options that can be applied to the calls, how to construct the APIs and how to filter responses. A good thing to remember is that some APIs change and get deprecated from version to version, and it's a good best practice to

for example, you can create documents in an index, update them, move them to another index, or remove them.

Elasticsearch Search API

As its name implies, these API calls can be used to query indexed data for specific information. Search APIs can be applied globally, across all available indices and types, or n an index. Responses will contain matches to the specific query.

API

ch API allows users to manage indices, mappings, and templates. use this API to create or delete a new index, check if a specific index e a new mapping for an index.

API

ific API calls that allow you to manage and monitor your lost of the APIs allow you to define which Elasticsearch node to call l node ID, its name or its address.

gins

Elasticsearch plugins are used to extend the basic Elasticsearch functionality in various, specific ways. There are types, for example, that add security functionality, discovery mechanisms, and analysis capabilities to Elasticsearch.

Regardless of what functionalities they add, Elasticsearch plugins belong to either of the following two categories: core plugins or community plugins. The former is supplied as part

```
1 cd /usr/share/elasticsearch
2 sudo bin/elasticsearch-plugin install x-pack
```

Plugins must be installed on every node in the cluster, and each node must be restarted after installation.

e a bit different as each of them has different installation instructions.

ins are installed the same way as core plugins but require additional ation steps.

earch, detailed some of its core concepts and explained the REST ng about Elasticsearch, here are some resources you may find

torial: Getting Started

sheet

ix the Top 5 Elasticsearch Mistakes

Logstash

What is Logstash?

In the ELK Stack (Elasticsearch, Logstash and Kibana), the crucial task of parsing data is given to the "L" in the stack – Logstash.

Logstash started out as an open source tool developed to handle the streaming of a large amount of log data from multiple sources. After being incorporated into the ELK Stack, it

ck's workhorse, in charge of also processing the log messages, ıassaging them and then dispatching them to a defined destination

ystem of plugins, Logstash can be used to collect, enrich and of different data types. There are over 200 different plugins for ommunity making use of its extensible features.

smooth sailing for Logstash. Due to some inherent performance s, Logstash has received a decent amount of complaints from users ojects were developed to alleviate some of these issues (e.g. Forwarder, Beats), and alternative log aggregators began competing

s, Logstash still remains a crucial component of the stack. Big steps have been made to try and alleviate these pains by introducing Beats and improvements to Logstash itself, ultimately make logging with ELK much more reliable than what it used to be.

Read more about installing and using Logstash in our Logstash tutorial.

support codecs that allow you to encode or decode your data (e.g. json, multiline, plain).

Input plugins

One of the things that makes Logstash so powerful is its ability to aggregate logs and events from various sources. Using more than 50 input plugins for different platforms, tions, Logstash can be defined to collect and process data from d them to other systems for storage and analysis.

uts used are: file, beats, syslog, http, tcp, udp, stdin, but you can of other sources.

umber of extremely powerful filter plugins that enable you to enrich, ss logs. It's the power of these filters that makes Logstash a very tool for parsing log data.

d with conditional statements to perform an action if a specific

uts used are: grok, date, mutate, drop. You can read more about gstash Filter Plugins.

Output plugins

As with the inputs, Logstash supports a number of output plugins that enable you to push your data to various locations, services, and technologies. You can store events using outputs such as File, CSV, and S3, convert them into messages with RabbitMQ and SQS, or

output plugins.

Logstash Codecs

Codecs can be used in both inputs and outputs. Input codecs provide a convenient way to decode your data before it enters the input. Output codecs provide a convenient way to re it leaves the output.

:

codec is for plain text with no delimitation between events

for encoding JSON events in inputs and decoding json messages in outputs
vert to plain text if the received payloads are not in a valid JSON format

dec allows you either to receive and encode json events delimited by \n or to
ages delimited by \n in outputs

hich is very useful in debugging, allows you to output Logstash events as

ample

configuration DSL that enables you to specify the inputs, outputs, and filters described above, along with their specific options. Order matters, specifically around filters and outputs, as the configuration is basically converted into code and then executed. Keep this in mind when you're writing your configs, and try to debug them.

Input

The input section in the configuration file defines the input plugin to use. Each plugin has

```
 9  }
10
11  }
```

Here we are using the file input plugin. We entered the path to the file we want to collect, [illegible] osition as beginning to process the logs from the beginning of the

configuration file defines what filter plugins we want to use, or in cessing we want to apply to the logs. Each plugin has its own which you should research before using.

```
" => "%{COMBINEDAPACHELOG}" }

 9  date {
10
11  match => [ "timestamp" , "dd/MMM/yyyy:HH:mm:ss Z" ]
12
13  }
14
15  geoip {
16
17  source => "clientip"
18
```

Output

The output section in the configuration file defines the destination to which we want to send the logs to. As before, each plugin has its own configuration options, which you should research before using.

Table of Contents


```
:9200"]
```

defining a locally installed instance of Elasticsearch.

he Logstash configuration file should look as follows:

```
13 filter {
14
15 grok {
16
17 match => { "message" => "%{COMBINEDAPACHELOG}" }
18
19 }
20
21 date {
                    mp" , "dd/MMM/yyyy:HH:mm:ss Z" ]
```

Table of Contents


```
)"

t:9200"]
```

Logstash pitfalls

As implied above, Logstash suffers from some inherent issues that are related to its design. Logstash requires JVM to run, and this dependency coupled with the implementation in Ruby became the root cause of significant memory consumption, especially when multiple pipelines and advanced filtering are involved.

- Add a buffer – a recommended method involves adding a queuing layer between Logstash and the destination. The most popular methods use Kafka, Redis and RabbitMQ.
- Persistent Queues – a built-in data resiliency feature in Logstash that allows you to store data in an internal queue on disk. Disabled by default — you need to enable the feature in the Logstash settings file.
- Dead Letter Queues – a mechanism for storing events that could not be processed on disk. — you need to enable the feature in the Logstash settings file.

Table of Contents

o look out for, refer to the 5 Logstash Pitfalls article.

ement in your ELK Stack, but you need to know how to use it both
d together with the other components in the stack. Below is a list of
ill help you use Logstash.

plugins

ns

h

- Kibana tutorial

Did we miss something? Did you find a mistake? We're relying on your feedback to keep this guide up-to-date. Please add your comments at the bottom of the page, or send them to: elk-guide@logz.io

Without being able to efficiently query and monitor data, there is little use to only aggregating and storing it. Kibana plays that role in the ELK Stack — a powerful analysis and visualization layer on top of Elasticsearch and Logstash.

What is Kibana?

Table of Contents

ce, Kibana is a browser-based user interface that can be used to sualize the data stored in Elasticsearch indices (Kibana cannot be th other databases). Kibana is especially renowned and popular due d visualization capabilities that allow users to explore large volumes

d on Linux, Windows and Mac using .zip or tar.gz, repositories or on n node.js, and the installation packages come built-in with the d more about setting up Kibana in our Kibana tutorial.

ges have been made in more recent versions to the licensing model, of basic X-Pack features into the default installation packages.

Searching Elasticsearch for specific log message or strings within these messages is the bread and butter of Kibana. In Kibana's query bar, you can enter Lucene query syntax or searches based on Elasticsearch Query DSL. Once entered, the main display area will filter the data displayed accordingly, showing matches in reverse chronological order.

Kibana querying is an art unto itself, and there are various methods for performing

Kibana searches cheat sheet

Below is a list of some tips and best practices for using the above-mentioned search types:

- Use free-text searches for quickly searching for a specific string. Use double quotes (“string”) to look for an exact match.
  Example: “USA“

ymbol to replace any number of characters and the ? wildcard symbol to
aracter.

efix for a field to search for logs that have that field.
esponse

nge within a field.
e brackets [], this means that the results are inclusive. If you use {}, this means
exclusive.

statements (e.g. AND, OR, TO) within a search, use capital letters. Example:
00]

lefine negative terms.
[400 TO 500] AND NOT response:404

are useful for searching terms within a specific character proximity. Example:
earch for all the terms that are within two changes from [categovi]. Proximity
f resources – use wisely!

- Field level search for non analyzed fields work differently than free text search.
  Example: If the field value is Error – searching for field:*rror will not return the right answer.
- If you don’t specify a logical operator, the default one is OR.
  Example: searching for Error Exception will run a search for Error OR Exception
- Using leading wildcards is a very expensive query and should be avoided when possible.

of different charts and graphs, you can slice and dice your data any way you want. You will find that you can do almost whatever you want with your data.

Creating visualizations, however, is now always straightforward and can take time. Key to making this process painless is knowing your data. The more you are acquainted with the different nooks and crannies in your data, the easier it is.

re built on top of Elasticsearch queries. Using Elasticsearch , average, min, mac, etc.), you can perform various processing isualizations depict trends in the data.

a are categorized into five different types of visualizations:

Heat Map, Horizontal Bar, Line, Pie, Vertical bar)

auge, Goal, Metric)

lap, Region Map)

n, Visual Builder)

rkdown, Tag Cloud)

describe the main function of each visualization and a usage example:

Vertical Bar Chart: Great for time series data and for splitting lines across fields	URLs over time
Pie Chart: Useful for displaying parts of a	Top 5 memory consuming system procs

relationships between two fields

Line Chart: are a simple way to show time series and are good for splitting lines to show anomalies	Average CPU over time by host
Data Table: Best way to split across om way	Top user, host, pod, container by usage
the status of a resholds you	Memory consumption limits
ation for displaying e number	No. of Docker containers run.
gion Map: Help add sion to IP-based	Geographic origin of web server requests.
uery Builder: ore advanced series data	Percentage of 500 errors over time
y to add a customized text or image-based visualization to your dashboard based on markdown syntax	Company logo or a description of a dashboard
Tag Cloud: Helps display groups of words sized by their importance	Countries sending requests to a web server

Dashboards are highly dynamic – they can be edited, shared, played around with, opened in different display modes, and more. Clicking on one field in a specific visualization within a dashboard, filters the entire dashboard accordingly (you will notice a filter added at the top of the page).

For more information and tips on creating a Kibana dashboard, see Creating the Perfect Kibana Dashboard

Table of Contents

arch index

ations, and dashboards saved in Kibana are called objects. These dedicated Elasticsearch index (.kibana) for debugging, sharing, ackup.

s soon as Kibana starts. You can change its name in the Kibana ndex contains the following documents, each containing their own

ıs

s

- Saved dashboards

What's next?

This article covered the functions you will most likely be using Kibana for, but there are plenty more tools to learn about and play around with. There are development tools such

Beats

Table of Contents

raditionally consisted of three main components — Elasticsearch, is now also used together with what is called “Beats” — a family of nt use cases. The advent of the different beats — Filebeat, t, Auditbeat, Heartbeat and Winlogbeat — gave birth to a new title : Stack”.

of open source log shippers that act as agents installed on the ır environment for collecting logs or metrics. Written in Go, these d to be lightweight in nature — they leave a small installation efficient, and function with no dependencies.

he different beats vary — log files in the case of Filebeat, network data in the case of Packetbeat, server metrics in the case of Metricbeat, and so forth. In addition to the beats developed and supported by Elastic, there is also a growing list of beats developed and contributed by the community.

Once collected, you can configure your beat to ship the data either directly into Elasticsearch or to Logstash for additional processing.

Packetbeat

A network packet analyzer, Packetbeat was the first beat introduced. Packetbeat captures network traffic between servers, and as such can be used for application and performance monitoring. Packetbeat can be installed on the server being monitored or on its own dedicated server.

to use Packetbeat here.

ps various system-level metrics for various systems and platforms.
at also supports internal modules for collecting statistics from
can configure the frequency by which Metricbeat collects the
ific metrics to collect using these modules and sub-settings called

to use Metricbeat here.

terest Windows sysadmins or engineers as it is a beat designed
ıg Windows Event logs. It can be used to analyze security events, updates installed, and so forth.

Read more about how to use Winlogbeat here.

Configuring beats

Beats configuration files are based on the YAML format with a dictionary containing a group of key-value pairs, but they can contain lists and strings, and various other data types. Most of the beats also include files with complete configuration examples, useful for learning the different configuration settings that can be used. Use it as a reference.

Table of Contents

ıt support modules — built-in configurations and Kibana objects for systems. Instead of configuring these two beats, these modules will pre-configured settings which work just fine in most cases but that d fine tune as you see fit.

che, Auditd, Icinga, Kafka, Logstash, MySQL, Nginx, PostgreSQL,

pache, HAProxy, MySQL, Nginx, PostgreSQL, Redis, System,

ample

uration example look like? Obviously, this differs according to the beat in question. Below, however, is an example of a Filebeat configuration that is using a single prospector for tracking Puppet server logs, a JSON directive for parsing, and a local Elasticsearch instance as the output destination.

```
13 json.keys_under_root: true
14
15 output.elasticsearch:
16
17 # Array of hosts to connect to.
18
19 hosts: ["localhost:9200"]
```

Table of Contents

st practices

own unique configuration file and configuration settings, and wn set of instructions. Still, there are some common configuration be outlined here to provide a solid general understanding.

s Filebeat, include full example configuration files (e.g, .full.yml). These files include long lists all the available configuration options.

emely sensitive. DO NOT use tabs when indenting your lines — only spaces. files for Beats are mostly built the same way, using two spaces for

se Sublime) to edit the file.

ter is used for defining new elements — be sure to preserve their e hierarchies between sub-constructs.

Additional information and tips are available in the Musings in YAML article.

What next?

Beats are a great and welcome addition to the ELK Stack, taking some of the load off Logstash and making data pipelines much more reliable as a result. Logstash is still a

Did we miss something? Did you find a mistake? We're relying on your feedback to keep this guide up-to-date. Please add your comments at the bottom of the page, or send them to: elk-guide@logz.io

ELK in Production

become a must-do action for any organization to resolve problems ations are running in a healthy manner. As such, log management e, a mission-critical system.

ooting a production issue or trying to identify a security hazard, the d running around the clock. Otherwise, you won't be able to e issues that arise — potentially resulting in performance e or security breach. A log analytics system that runs continuously :ation with the means to track and locate the specific issues that are ur system.

In this article, I will share our experiences in building Logz.io. I will introduce some of the challenges and offer some related guidelines in building a production-grade ELK deployment.

1. Generally speaking, a production-grade ELK implementation needs to: Save and index all of the log files that it receives (sounds obvious, right?)

If you are troubleshooting an issue and go over a set of events, it only takes one missing logline to get incorrect results. Every log event must be captured. For example, you're viewing a set of events in MySQL that ends with a database exception. If you lose one of these events, it might be impossible to pinpoint the cause of the problem.

The recommended method to ensure a resilient data pipeline is to place a buffer in front of

entry point for all log events that are shipped to your system. It will
ıtil the downstream components have enough resources to index.

edis or Kafka, though also RabbitMQ can be used in this context.

gine at the heart of ELK. It is very susceptible to load, which means
ely careful when indexing and increasing your amount of
ticsearch is busy, Logstash works slower than normal — which is
es into the picture, accumulating more documents that can then be
h. This is critical not to lose log events.

ı/Elasticsearch Exceptions

ı trying to index logs in Elasticsearch that cannot fit into the
ed mapping.

For example, let's say you have a log entry that looks like this:

```
1 timestamp=time, type=my_app, error=3,….
```

But later, your system generates a similar log that looks as follows:

change is required, just change the index according to the type of log.

2. Ensure that Logstash is consistently fed with information and monitor Elasticsearch exceptions to ensure that logs are not shipped in the wrong formats. Using mapping that is fixed and less dynamic is probably the only solid solution here (that doesn't require you to start coding).

At Logz.io, we solve this problem by building a pipeline to handle mapping exceptions that eventually index these documents in manners that don't collide with existing mapping.

Table of Contents

wth and bursts

eeds and grows, so does your data. Machines pile up, environments ollow suit. As you scale out with more products, applications, nd operations, you also accumulate more logs. This requires a pute resource and storage capacity so that your system can process

ement solutions consume large amounts of CPU, memory, and re bursty by nature, and sporadic bursts are typical. If a file is base, the frequency of logs that you receive may range from 100 to er second.

o allocate up to 10 times more capacity than normal. When there is a real production issue, many systems generally report failures or disconnections, which cause them to generate many more logs. This is actually when log management systems are needed more than ever.

ELK Elasticity

datacenter — we recommend having a cluster of Elasticsearch nodes that run in different availability zones, or in different segments of a data center, to ensure high availability.

Let's take a look at some of the components required for a scalable ELK deployment.

Kafka

olacing a buffer in front of your indexing mechanism is critical to ents. It could be mapping conflicts, upgrade issues, hardware issues the volume of logs. Whatever the cause you need an overflow here Kafka comes into the picture.

ogs that are to be indexed, Kafka must persist your logs in at least 2 tain your data (even if it was consumed already by Logstash) for at

ning for the local storage available to Kafka, as well as the network the Kafka brokers. Remember to take into account huge spikes in ns of times more than "normal"), as these are the cases where you most.

anpower you will have to dedicate to fixing issues in your inning the retention capacity in Kafka.

Another important consideration is the ZooKeeper management cluster – it has its own requirements. Do not overlook the disk performance requirements for ZooKeeper, as well as the availability of that cluster. Use a three or five node cluster, spread across racks/availability zones (but not regions).

When considering consumption from Kafka and indexing you should consider what level of parallelism you need to implement (after all, Logstash is not very fast). This is important to understand the consumption paradigm and plan the number of partitions you are using in your Kafka topics accordingly.

Logstash

gstash instances to run is an art unto itself and the answer depends
tors: volume of data, number of pipelines, size of your Elasticsearch
epted latency — to name just a few.
uing mechanism with different scalable workers. When a queue is
nal workers to read into Elasticsearch.
ed the number of Logstash instances required, run each one of them
VS). This comes at a cost due to data transfer but will guarantee a
eline.
ate Logstash and Elasticsearch by using different machines for them.
they both run as JVMs and consume large amounts of memory,
ible to run on the same machine effectively.
out it is recommended allocating a maximum of 30 GB or half of the
ine for Logstash. In some scenarios, however, making room for
also a good best practice.

Elasticsearch cluster

Elasticsearch is composed of a number of different node types, two of which are the most important: the master nodes and the data nodes. The master nodes are responsible for cluster management while the data nodes, as the name suggests, are in charge of the data (read more about setting up an Elasticsearch cluster here).

machines with very fast storage and a large capacity for memory.

Run in Different AZs (But Not in Different Regions)

We recommend having your Elasticsearch nodes run in different availability zones or in different segments of a data center to ensure high availability. This can be done through an Elasticsearch setting that allows you to configure every document to be replicated . As with Logstash, the resulting costs resulting from this kind of ite steep due to data transfer.

s may contain sensitive data, it is crucial to protect who can see it access to specific dashboards, visualizations, or data inside your There is no simple way to do this in the ELK Stack.

e nginx reverse proxy to access your Kibana dashboard, which configuration that requires those who want to access the dashboard d password. This quickly blocks access to your Kibana console.

ses if you would like to limit access on a more granular level. This is d out-of-the-box within open source ELK. There are some open source solutions that can help (e.g. SearchGuard) or you can use Elastic's X-Pack and build the security from Elasticsearch up the stack. At Logz.io, we take a different approach that allows for role-based access.

Last but not least, be careful when exposing Elasticsearch because it is very susceptible to attacks. There are some basic steps to take that will help you secure your Elasticsearch

apply the relevant parsing abilities to Logstash — which has proven to be quite a challenge, particularly when it comes to building groks, debugging them, and actually parsing logs to have the relevant fields for Elasticsearch and Kibana.

At the end of the day, it is very easy to make mistakes using Logstash, which is why you should carefully test and maintain all of your log configurations by means of version

Table of Contents

e you may get started using nginx and MySQL, you may incorporate
you grow that result in large and hard-to-manage log files. The
ted a lot of solutions around this topic, but trial and error are
th open source tools before using them in production.

ntainability comes into play with excess indices. Depending on how
data, you need to have a process set up that will automatically
herwise, you will be left with too much data and your Elasticsearch
data loss.

appening, you can use Elasticsearch Curator to delete indices. We
ron job that automatically spawns Curator with the relevant
ny old indices, ensuring you don't end up holding too much data. It
o save logs to S3 in a bucket for compliance, so you want to be sure to have a copy of the logs in their original format.

Upgrades

Major versions of the stack are released quite frequently, with great new features but also breaking changes. It is always wise to read and do research on what these changes mean

between Logstash and Elasticsearch and breaking changes.

Kibana upgrades can be problematic, especially if you're running on an older version. Importing objects is "generally" supported, but you should backup your objects and test the upgrade process before upgrading in production. As always — study breaking changes!

Table of Contents

K to process logs from a server or two is easy and fun. Like any
m, it takes much more work to reach a solid production deployment.
we've been working with many users who struggle with making ELK
on. Read more about the real cost of doing ELK on your own.

g? Did you find a mistake? We're relying on your feedback to keep
Please add your comments at the bottom of the page, or send them

Common Pitfalls

Like any piece of software, the ELK Stack is not without its pitfalls. While relatively easy to set up, the different components in the stack can become difficult to handle as soon as you

basic configurations, others are related to best practices. In this section of the guide, we will outline some of these mistakes and how you can avoid making them.

Elasticsearch

ırch mapping

.icsearch, create an index, and feed it with JSON documents without . Elasticsearch will then iterate over each indexed field of the JSON field, and create a respective mapping. While this may seem ideal, s are not always accurate. If, for example, the wrong field type is errors will pop up.

ıould define mappings, especially in production-line environments. ıdex a few documents, let Elasticsearch guess the field, and then eates with GET /index_name/doc_type/_mapping. You can then take hands and make any appropriate changes that you see fit without chance.

ex your first document like this:

```
1 {
2
3 "action": "Some action",
4
5 "payload": "2016-01-20"
6
7 }
```

Elasticsearch will automatically map the “payload” field as a date field

new index will not be saved because Elasticsearch has already marked it as "date".

Capacity Provisioning

Provisioning can help to equip and optimize Elasticsearch for operational performance. It requires that Elasticsearch is designed in such a way that will keep nodes up, stop memory from growing out of control, and prevent unexpected actions from shutting down nodes.

need?" is a question that users often ask themselves. Unfortunately, , but certain steps can be taken to assist with the planning of

ual use-case. Boot up your nodes, fill them with real documents, and ard breaks.

mind that the concept of "start big and scale down" can save you compared to the alternative of adding and configuring new nodes ount is no longer enough.

rd's capacity, you can easily apply it throughout your entire index. It derstand resource utilization during the testing process because it he proper amount of RAM for nodes, configure your JVM heap our overall testing process.

Oversized Template

Large templates are directly related to large mappings. In other words, if you create a large mapping for Elasticsearch, you will have issues with syncing it across your nodes, even if you apply them as an index template.

By default, the first cluster that Elasticsearch starts is called elasticsearch. If you are unsure about how to change a configuration, it's best to stick to the default configuration. However, it is a good practice to rename your production cluster to prevent unwanted nodes from joining your cluster.

Below is an example of how you might want to rename your cluster and nodes:


```
.csearch_production
earch_node_001
```

file

pain points not only for working with Logstash but for the entire re ELK-based pipelines stalled because of a bad Logstash ot an uncommon occurrence.

plugins with their own options and syntax instructions, differently iles, files that tend to become complex and difficult to understand just some of the reasons why Logstash configuration files are the peline.

As a rule of the thumb, try and keep your Logstash configuration file as simple as possible. This also affects performance. Use only the plugins you are sure you need. This is especially true of the various filter plugins which tend to add up necessarily.

If possible — test and verify your configurations before starting Logstash in production. If you're running Logstash from the command line, use the –config.test_and_exit parameter

Recent versions of Logstash and the ELK Stack have improved this inherent weakness. Using Filebeat and/or Elasticsearch Ingest Node, some of the processing can be outsourced to the other components in the stack. You can also make use of monitoring APIs to identify bottlenecks and problematic processing.

Slow processing

ces, a complex or faulty configuration file, or logs not suiting the lt in extremely slow processing by Logstash that might result in data

onitor key system metrics to make sure you're keeping tabs on - monitor the host's CPU, I/O, memory and JVM heap. Be ready to configurations accordingly (e.g. raising the JVM heap size or raising workers). There is a nice performance checklist here.

ug-in that extracts keys and values from a single log using them to e structured data format. For example, let's say a logline contains hrough a key-value filter, it will create a new field in the output JSON vould be "x" and the value would be "5".

By default, the key-value filter will extract every key=value pattern in the source field. However, the downside is that you don't have control over the keys and values that are created when you let it work automatically, out-of-the-box with the default configuration. It may create many keys and values with an undesired structure, and even malformed keys that make the output unpredictable. If this happens, Elasticsearch may fail to index the resulting document and parse irrelevant information.

make sure the two behave nicely together.

Defining an index pattern

There's little use for of an analysis tool if there is no data for it to analyze. If you have no data indexed in Elasticsearch or have not defined the correct index pattern for Kibana to [illegible] 's work cannot start.

lata pipeline is working as expected and indexing data in
do this by querying Elasticsearch indices), and b) you have defined
rn in Kibana (Management → Index Patterns in Kibana).

sticsearch

Table of Contents

As the message reads, Kibana simply cannot connect to an Elasticsearch instance. There are some simple reasons for this — Elasticsearch may not be running, or Kibana might be configured to look for an Elasticsearch instance on a wrong host and port.

available. From free-text searches to field-level and regex searches, there are many options, and this variety is one of the reasons that people opt for the ELK Stack in the first place. As implied in the opening statement above, some Kibana searches are going to crash Elasticsearch in certain circumstances.

For example, using a leading wildcard search on a large dataset has the potential of
l should, therefore, be avoided.

ldcard queries if possible, especially when performed against very

configurations can cause your browser to crash. For example, owser and system settings, changing the value of the etting to a high number can easily cause Kibana to freeze.

olks at Elastic have placed a warning at the top of the page that is us to be extra careful. Anyone with a guess on how successful this

Table of Contents

Beats

The log shippers belonging to the Beats family are pretty resilient and fault-tolerant. They were designed to be lightweight in nature and with a low resource footprint.

YAML configuration files

Filebeat is an extremely lightweight shipper with a small footprint, and while it is extremely rare to find complaints about Filebeat, there are some cases where you might run into high CPU usage.

One factor that affects the amount of computation power used is the scanning frequency ich Filebeat is configured to scan for files. This frequency can be ector using the scan_frequency setting in your Filebeat you have a large number of prospectors running with a tight scan sult in excessive CPU usage.

remember the previous reading for each log file being harvested by elps Filebeat ensure that logs are not lost if, for example, ash suddenly go offline (that never happens, right?).

o your local disk in a dedicated registry file, and under certain reating a large number of new log files, for example, this registry file e and begin to consume too much memory.

at there are some good options for making sure you don't fall into use the *clean_removed* option, for example, to tell Filebeat to clean non-existing files from the registry file.

Filebeat – Removed or Renamed Log Files

File handlers for removed or renamed log files might exhaust disk space. As long as a harvester is open, the file handler is kept running. Meaning that if a file is removed or

Summing it up

The ELK Stack is a fantastic piece of software with some known and some less-known weak spots.

The good news is that all of the issues listed above can be easily mitigated and avoided as described. The bad news is that there are additional pitfalls that have not been detailed

with more tips and best practices to help avoid them:

Mistakes

ou Need to Avoid

Be Aware Of

sh Elasticsearch

search.

g? Did you find a mistake? We're relying on your feedback to keep Please add your comments at the bottom of the page, or send them

Use Cases

affects almost all the steps implemented along the way — where and how to install the stack, how to configure your Elasticsearch cluster and which resources to allocate to it, how to build data pipelines, how to secure the installation — the list is endless.

So, what are you going to be using ELK for?

d troubleshooting

being in handy during a crisis. The first place one looks at when an our error logs and exceptions. Yet, logs come in handy much earlier ycle.

s in log-driven development, where logging starts from the very first en subsequently instrumented throughout the entire application. into your code adds a measure of observability into your in handy when troubleshooting issues.

oping a monolith or microservices, the ELK Stack comes into the eans for developers to correlate, identify and troubleshoot errors place, preferably in testing or staging, and before the code goes a variety of different appenders, frameworks, libraries and shippers, log messages are pushed into the ELK Stack for centralized management and analysis.

Once in production, Kibana dashboards are used for monitoring the general health of applications and specific services. Should an issue take place, and if logging was instrumented in a structured way, having all the log data in one centralized location helps make analysis and troubleshooting a more efficient and speedy process.

considerations: how to access each machine, how to collect the data, how to add context to the data and process it, where to store the data and how long to store it for, how to analyze the data, how to secure the data and how to back it up.

The ELK Stack helps by providing organizations with the means to tackle these questions by providing an almost all-in-one solution. Beats can be deployed on machines to act as data to Logstash instances. Logstash can be configured to d process it before indexing the data in Elasticsearch. Kibana is then ta, detect anomalies, perform root cause analysis, and build ashboards.

/hile Elasticsearch was initially designed for full-text search and gly being used for metrics analysis as well. Monitoring performance onent in your architecture is key for gaining visibility into operations. :s can be done using 3rd party auditing or monitoring agents or e available beats (e.g. Metricbeat, Packetbeat) and Kibana now ships types to help analyze time series (Timelion, Visual Builder).

ıpliance

en crucial for organizations. Yet over the past few years, because of both an increase in the frequency of attacks and compliance requirements (HIPAA, PCI, SOC, FISMA, etc.), employing security mechanisms and standards has become a top priority.

Because log data contains a wealth of valuable information on what is actually happening in real time within running processes, it should come as little surprise that security is fast becoming a strong use case for the ELK Stack.

1.Anti-DDoS

Once a DDoS attack is mounted, time is of the essence. Quick identification is key to minimizing the damage, and that's where log monitoring comes into the picture. Logs contain the raw footprint generated by running processes and thus offer a wealth of information about what is happening in real time.

rganizations can build a system that aggregates data from the environment (web server, databases, firewalls, etc.), process the and visualizes the data in powerful monitoring dashboards.

enterprise security management that seeks to provide a holistic n's IT security. The main purpose of SIEM is to provide a prehensive view of your IT security. The SIEM approach includes a rd that allows you to identify activity, trends, and patterns easily. If , SIEM can prevent legitimate threats by identifying them early, ity, providing compliance reports, and supporting incident-response

instrumental in achieving SIEM. Take an AWS-based environment as an example. Organizations using AWS services have a large amount of auditing and logging tools that generate log data, auditing information and details on changes made to the configuration of the service. These distributed data sources can be tapped and used together to give a good and centralized security overview of the stack.

Read more about SIEM and ELK here.

databases, supply chains, personnel records, manufacturing data, sales and marketing campaigns, and more. The data itself might be stored in internal data warehouses, private clouds or public clouds, and the engineering involved in extracting and processing the data (ETL) has given rise to a number of technologies, both proprietary and open source. As with the previous use cases outlined here, the ELK Stack comes in handy for pulling data from these varied data sources into one centralized location for analysis. For example,

ver access logs to learn how our users are accessing our website, CRM system to learn more about our leads and users, or we might marketing automation tool provides.

ch of proprietary tools used for precisely this purpose. But the ELK open source option to perform almost all of the actions these tools

ner edge use case for the ELK Stack but a relevant one nonetheless. ith ELK? Well, the common denominator is of course logs.

s (Apache, nginx, IIS) reflect an accurate picture of who is sending te, including requests made by bots belonging to search engines experts will be using this data to monitor the number of requests made by Baidu, BingBot, GoogleBot, Yahoo, Yandex and others.

Technical SEO experts use log data to monitor when bots last crawled the site but also to optimize crawl budget, website errors and faulty redirects, crawl priority, duplicate crawling, and plenty more. Check out our guide on how to use log data for technical SEO.

method you choose will depend on your requirements, specific environment, preferred toolkit, and many more.

Over the last few years, we have written a large number of articles describing different ways to integrate the ELK Stack with different systems, applications and platforms. The method varies from a data source to data source — it could be a Docker container, Filebeat ash and so forth. Just take your pick.

Table of Contents

integrations just in case you're looking into implementing it. We've n into separate categories for easier navigation.

include Logz.io-specific instructions as well, including ready-made art of our ELK Apps library. Integrations with instructions for gz.io ELK are marked.

Web servers

- Apache
- Nginx
- IIS

- Heroku*

Databases

- MySQL*
- MongoDB

Table of Contents

Docker

- Docker logging with ELK – Part 1
- Docker logging with ELK – Part 2

Azure

- Network Security Group Flow logs

Security

- Wazuh

: for SIEM

Table of Contents

Additional Resources

General

- 10 Resources to Bookmark if You're Running ELK
- Elastic Stack 6 – What You Need to Know
- The Cost of Doing ELK on Your Own

- A Beginner's Guide to Logstash Grok
- Monitoring Logstash Pipelines
- Fluentd vs. Logstash
- A Guide to Logstash Plugins

Table of Contents

t Kibana Dashboard

ɔana Visualizations

Related Posts

What's New in Kibana 6.3

Table of Contents

Table of Contents