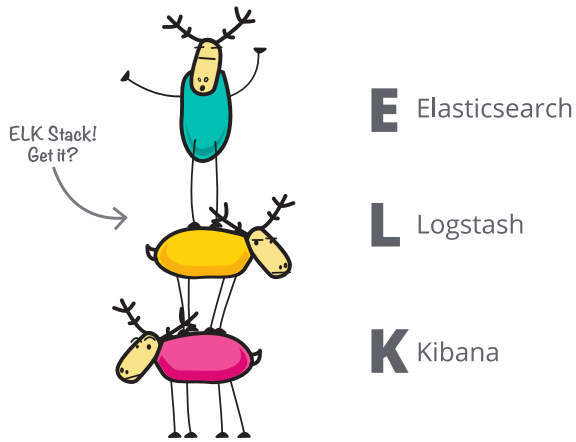
becon
we drive digital innovation

Mit dem Elch Logging zentralisieren & Servermetriken visualisieren

Wenn man in größeren IT-Landschaften viele Dienste und Funktionen hat, bekommt man zwangsläufig auch viele Logdateien und Statusmeldungen in unterschiedlichen Formaten. Diese Informationen liegen auch oft noch verteilt und lokal auf den entsprechenden Systemen. Das erschwert natürlich die schnelle und effiziente Auswertung dieser Log- und Statusmeldungen. Dadurch sind Fehler oder plötzlich auftretende Probleme schwer zu erkennen oder zu analysieren. Auch das proaktive Handeln für bestimmte Situationen ist so gut wie unmöglich. Man kann von Administratoren nicht erwarten, sich auf mehrere Server einzuloggen und verschiedene Logdateien per „grep" zu durchsuchen, um z.B. eine DOS-Attacke zu erkennen und abzuwehren. Oder wenn man wissen will, wer sich immer wieder mit einem falschen Account versucht an einem Switch oder Server anzumelden. Es gibt eine Menge Beispiele warum man Log- und Systemmeldungen auswerten sollte.
Da ist das Konzentrieren und Zentralisieren dieser Log- und Statusinformationen hilfreich. Dazu gibt es mehrere Wege. Wer den Open-Source-Weg gehen möchte, sollte sich den ELK-Stack (bzw. den Elastic-Stack) anschauen.

Was ist nun eigentlich ELK?

ELK bzw. der ELK-Stack ist eine Kombination aus Elasticsearch, Logstash und Kibana. Und wenn man es genau nimmt, muss man heutzutage noch ein B für **Beats** einbauen. Aber dazu später mehr.



Sehen wir uns mal kurz die einzelnen Komponenten der Elchfamilie an:

Logstash ist ein Tool, um die anfallenden Daten zu sammeln, zu indexieren, zu transformieren, zu filtern und vieles mehr. Diese Daten werden anschließend zur Verarbeitung weiter gegeben. In diesem Fall an Elasticsearch. Logstash kann viele Formate als Input verarbeiten und auch in alle möglichen Formate als Output weiter geben.

Elasticsearch ist eine NoSql-Datenbank mit einer sehr performanten Searchengine, mit der man vorher definierte Ergebnisse auch aus großen Datenmengen in (fast) Echtzeit erhält.

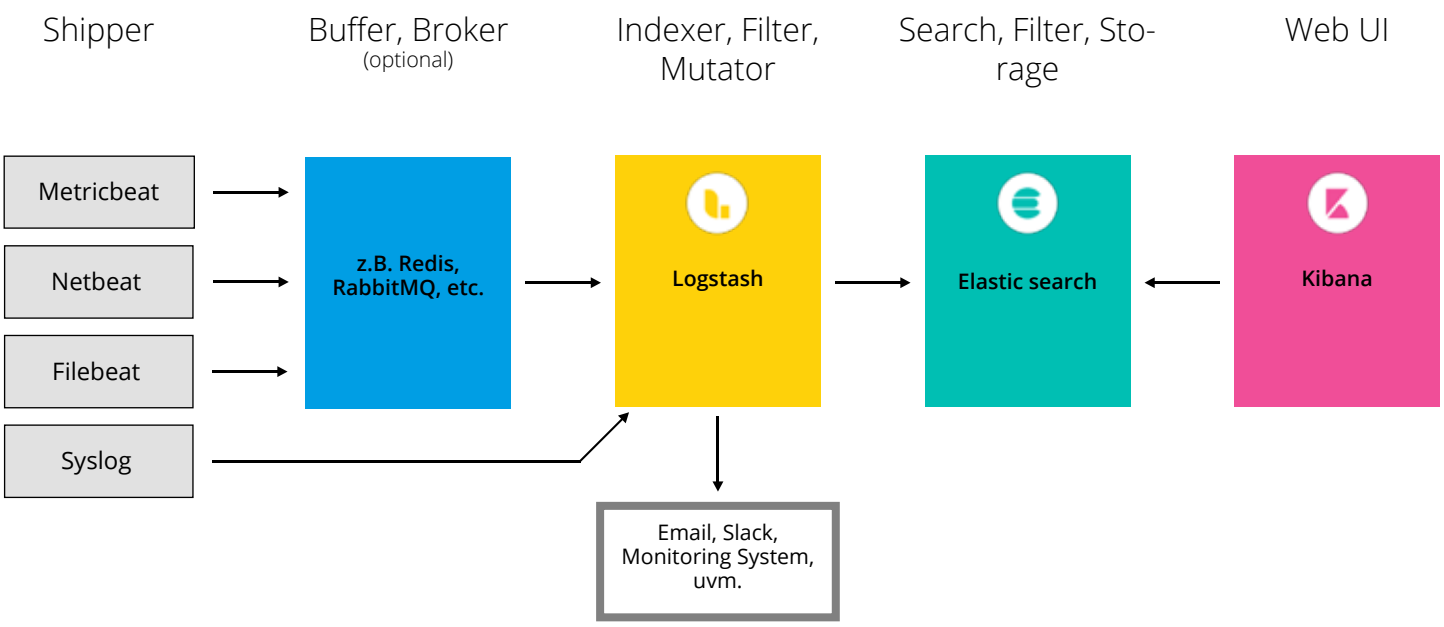
Kibana ist ein Webfrontend, um die Daten aus Elasticsearch bequem und schnell zu visualisieren. Das geht von einer einfachen Textdarstellung bis hin zu eigenen Dashboards, Diagrammen und anderen Anzeigen.

Und dann gibt es, wie schon erwähnt, auch noch **Beats**. Das ist eine Sammlung sogenannter Shipper, also kleiner Agenten, die die Daten von verschiedenen Systemen an Logstash und/oder Elasticsearch schicken.

Diese Tools stehen für Linux, MacOS und für Windows zur Verfügung. Shipper können aber auch andere Datenquellen sein. Zum Beispiel Syslog, SNMP, Textfiles und vieles, vieles mehr. Den ELK-Stack mit Beats bezeichnet man dann als Elastic-Stack.

Optional kann man einen Puffer bzw. Broker zwischen Shipper und Logstash konfigurieren. Das hat den Vorteil, dass keine Daten verloren gehen, wenn man z.B. an der Konfiguration dahinter etwas ändert und man den Dienst neu starten muss oder z.B. das Netzwerk oder der Serververband, in dem sich der ELK-Stack befindet, kurzzeitig nicht erreichbar ist. Hier hat sich Redis als sehr gut funktionierende Lösung erwiesen. Aber auch andere Programme wie RabbitMQ eignen sich dafür.

Im Prinzip funktioniert das Ganze so:



Es ist aber nicht zwingend notwendig, dass die Shipper ihre Daten an Logstash direkt liefern, um dort aufgearbeitet zu werden. Man könnte die Daten auch direkt an Elasticsearch senden. Dann hat man aber meist unstrukturierte Daten in der Datanbank und das erschwert beispielsweise die Suche. Ein großer Vorteil von Logstash ist auch die Formatvielfalt, die es als Input und Output bedienen kann. Um die Daten für Elasticsearch „mundgerecht" aufzuarbeiten, gibt es unter anderem das Plugin **grok**. grok teilt z.B. eine Logmeldung in einzelne Felder auf, kann aber auch neue Felder hinzufügen. Diese Felder sind für Elasticsearch wesentlich schneller abzuarbeiten.

grok macht z.B. aus dieser Meldung:
Jan 1 06:25:43 mailserver14 postfix/cleanup [21403]:BEF25A72965: message-id=<20130101142543.5828399CCAF@mailserver14.example.com>

diese Felder:
- timestamp: Jan 1 06:25:43
- logsource: mailserver14
- program: postfix/cleanup
- pid: 21403
- queue_id: BEF25A72965
- syslog_message:messageid<20130101142543.5828399CCAF@mailserver14.example.com>

Elasticsearch kann mit diesen Feldern wesentlich effizienter arbeiten.

Bei bestimmten Ereignissen, die man vorher definiert hat, kann Logstash auch direkt ein entsprechendes Event erzeugen bzw. eine Meldung versenden. So kann man per Email und/oder Slack benachrichtigt werden und es erscheint zeitgleich ein entsprechendes Pop-up im Monitoring-System. Eine tolle Ergänzung zu bestehenden Monitoring-Systemen.

Möchte man nun die aufgearbeiteten Daten nach bestimmten Gesichtpunkten betrachten oder durchsuchen, eignet sich hierfür Elasticsearch. Am Schnellsten geht das mit dem WebFrontend Kibana, dass extra dafür entwickelt wurde, um eine visuelle Oberfläche für Elasticsearch zu sein. Je nach Anwendungsfall kann man sich schnell durch die entsprechenden Logdateien, Statusmeldungen, etc. arbeiten oder sich kleine Widgets und Dashboards bauen, um schnell einen Überblick über bestimmte Ereignisse zu haben. Elasticsearch kann natürlich viel mehr als „nur" Logdateien zu durchsuchen. Aber das würde hier den Rahmen sprengen.

Gerne helfen wir Ihnen bei der Implementierung, begleiten, supporten und schulen Sie. Kontaktieren Sie uns ganz unverbindlich. Wir freuen uns von Ihnen zu hören!

Kontakt:

becon GmbH
Hauptstraße 8b
82008 Unterhaching

T.: +49 89 608668-0
info@becon.de

www.becon.de