Unter **Logmanagement** werden hier alle Maßnahmen verstanden, die die Protokollierung und Auswertung von Laufzeitinformationen zu einem System oder zu einer Anwendung betreffen. Die protokollierten Daten sollen das System- oder Anwendungsverhalten z.B. aus Betriebs- oder Datenschutzsicht nachvollziehbar und analysierbar machen.

In der Vergangenheit wurden Auffälligkeiten, die z.B. über das Monitoring erkannt wurden, analysiert, indem ein Entwickler oder Administrator sich auf den jeweiligen Rechner geschaltet und einen Blick in die zugehörige Logdatei geworfen hat, um über Textsuche - z.B. über die Zeitstempel - Aufschluss über die Machenschaften des Systems zu bekommen.

Heutzutage sind die Gegebenheiten meistens etwas komplexer. Man nehme z.B. eine eCommerce-Plattform. Auf der Backend-Seite sind zu einem Geschäftsvorgang mehrere Systeme wie Load Balancer, Reverse Proxies, Application Server etc. involviert. All diese Systeme, die noch dazu auf unterschiedlichen virtuellen Servern gehostet sind, werden zur Ausführung des Vorgangs durchlaufen, und jedes dieser Systeme protokolliert in eigene Logdateien. Um einen Geschäftsvorgang lückenlos nachvollziehen zu können, sind somit unterschiedliche Logfiles mit möglicherweise unterschiedlichen Formaten auf verschiedenen Rechnern zu analysieren. Man kann an dem Beispiel sehen, dass es sinnvoll ist, Daten auf einem Logserver zu zentralisieren, also von verschiedenen Systemen zusammenzuführen und miteinander zu korrellieren, um so an einer zentralen Stelle Analysen und Auswertungen vornehmen zu können.

Im Umfeld von virtuellen Maschinen, Clustering, Cloud oder IoT ist ein zentrales Logmanagement aus folgenden Gründen sinnvoll:

- **Virtuelle Maschinen** können z.B. im Continuous-Integration-Umfeld schnell aufgesetzt und gelöscht werden. Ohne zentrales Logmanagement gehen diese Logs verloren
- **Verteilte Systeme** und heterogene Systemlandschaften bedeuten, dass ein Vorgang über verschiedene Knoten hinweg verfolgt werden muss. Dazu sind für jeden Knoten Zugriffsrechte erforderlich, an jedem Knoten neu anmelden, an jedem Knoten ein Logfile analysieren.
- **Cloud**

Typische Fragestellungen für die Analyse der Logdaten sind z.B.:

- Warum laufen die Regressionstests zu einem Build nicht fehlerfrei durch?
- Was ist die Ursache für den im Monitoringsystem gemeldeten Fehler?
- Warum wurde die Geschäftstransaktion abgebrochen?
- Ist der Traffic des Load Balancer ok?
- Haben die aktuellen Performance-Probleme mit dem Load-Balancer zu tun?

Typische Probleme bei der Auswertung der Logs:

- Über welche Systeme sind die Logs verteilt?
- Habe ich darauf Zugriff?
- Wie korreliere ich die Logs?
- Wie ist das Format der Logdatei?

Eine zentrale Log-Management-Lösung setzt sich aus folgenden Abläufen zusammen:

- Einsammeln der Daten von den verschiedenen Quellen mit unterschiedlichen Protokollen und Datenformaten
- Interpretation und Normalisierung der Logdaten
- Idexierung der Daten
- Benutzerschnittstelle für das Absetzen von Such- und Filterabfragen und das Anzeigen von Suchergebnissen

Ein zentrales Logmanagement kann somit

- Geschäftsvorgänge über Maschinen hinweg verfolgen,
- Fehleranalysen über Maschinen hinweg durchführen,
- die Daten zu einem Geschäftsvorgang über verschiedene Prozesse und Maschinen hinweg korrelieren.

Für ein zentrales Logmanagement gibt es unterschiedliche Lösungsansätze - vom ELK-Stack über Apache Flume bis hin zu einer „reliable logging pipeline based on Kafka and Spark“.

Ein aktuell gängiger Ansatz basiert auf dem ELK-Stack, der aus folgenden Komponenten besteht:

- **Elasticsearch** als Volltextserver,
- **Logstash** für die Entgegennahme und die Aufbereitung der Daten,
- **Kibana** für die webbasierte Visualisierung in Form von Dashboards.

Welche Logtechnik die geeignetste ist, hängt von den technischen Rahmenbedinungen ab und muss entsprechend analysiert werden. Dass ein zentrales Logmanagement sinnvoll ist, sollte der Artikel veranschaulicht haben.