Lassen sich der gelbe Elefant und seine Freunde überhaupt noch beherrschen?

Aufbau und Betrieb einer Private-Analytics-Plattform

Ein Beitrag von
Fabian Hardt

Manuel, Systemadministrator in einem mittelständischen kunststoffverarbeitenden Betrieb, sitzt um 20:00 Uhr noch an seinem Arbeitsplatz und flucht leise vor sich hin. Er verwaltet neuerdings die Hadoop-Plattform, die ein Beratungshaus im vergangenen Jahr für seine Firma aufgesetzt hat. Massendatenverarbeitung von Sensorwerten aus der Produktion sind das Ziel dieser neuartigen Umgebung, doch bereits jetzt sind die Endanwender unzufrieden, weil es immer wieder zu Problemen im täglichen Betrieb kommt.

Sonderdruck aus BI-SPEKTRUM 4/2021

Aktuell arbeitet Manuel daran, die SSL-Zertifikate diverser Hadoop Services zu tauschen, da diese teilweise bereits abgelaufen sind. Leider gab es hierfür bislang kein geeignetes Monitoring, sodass er diese kritische Aktivität nicht kommen sehen konnte. Doch der größte Schmerzpunkt von Manuel liegt in der fehlenden Automation dieser Aufgabenstellung, da er die neuen SSL-Zertifikate nun händisch erzeugen und anschließend in verschiedenen Ordnern auf verschiedenen Servern ablegen muss. Um diese Stellen zu identifizieren, arbeitet sich Manuel schon seit heute Morgen durch die seitenlange Dokumentation des Herstellers, die leider immer wieder Lücken aufweist. Eigentlich wünscht er sich, dass sein Unternehmen die Analytics-Plattform in einer Public Cloud betreiben würde, dann könnte er heute Abend einfach sein Fußballtraining mit den Jungs genießen.

Geht es Ihnen ähnlich wie Manuel? Haben auch Sie schlechte Erfahrungen mit Hadoop gemacht und Ihre On-Premises-Analytics-Plattform ist schwer aufzubauen und zu administrieren? In diesem Artikel soll aufgezeigt werden, wie sich die administrativen Herausforderungen durch einen hohen Grad an Automatisierung sowie ein flächendeckendes Monitoring verringern lassen.

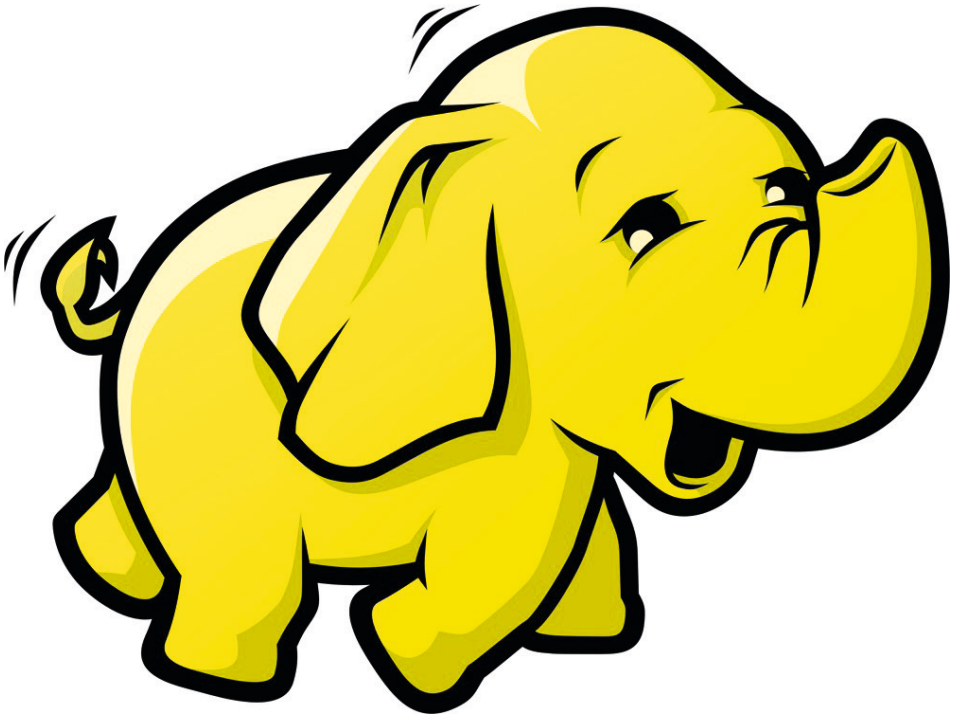

Bild: The Apache Software Foundation

Public vs. Private Cloud

Anhand des eingangs genannten Szenarios wird schnell deutlich, dass ein hoher Grad an Automatisierung sowohl beim Aufbau als auch beim Betrieb einer modernen Analytics-Plattform sehr wichtig ist [Ste20]. Diesen Vorteil bringen Public Clouds bereits von Hause aus mit, indem sie Software-as-a-Service-(SaaS-)Angebote bereitstellen, die mit wenigen Klicks und in kurzer Zeit aktiviert werden können. Darüber hinaus wird der Betrieb vollständig vom Cloud-Anbieter übernommen.

Um Manuels wiederkehrende „Handarbeit“ zu reduzieren und ähnliche Vorteile in einer On-Premises-Umgebung zu erreichen, muss auf Automatisierungs-Tools wie Ansible oder Terraform zurückgegriffen werden. Mit diesen Werkzeugen lassen sich auch im eigenen Rechenzentrum eine einfache Installation von Komponenten sowie ein stringentes Konfigurationsmanagement erreichen. Diese automatische Erzeugung von SSL-Zertifikaten sowie die anschließende Verteilung hätten auch direkt bei der Installation des Hadoop-Clusters automatisiert werden können (vgl. Abbildung 1), sodass Manuel die Zertifikate per Knopfdruck hätte verlängern können. Als gut geeignete Public-Key-Infrastruktur (PKI) kommt beispielsweise ein Hashicorp Vault Server in Frage, der sehr vielfältige APIs sowie eine ausgereifte Ansible Integration anbietet. Insbesondere bei wiederkehrenden, administrativen Aufgaben ist dieses Vorgehen dringend zu empfehlen.

Das Gleiche gilt für das Thema Skalierung. Hier tut sich die Public Cloud verständlicherweise sehr viel leichter, da die zur Verfügung stehende Rechenkapazität wesentlich größer ist und daher in Summe über viele Kunden und Systeme hinweg deutlich effizienter genutzt werden kann. Außerdem stehen stets ausreichend freie Reserven zur Verfügung, um eine nachträgliche Skalierung der Systeme zu ermöglichen [Sha17]. Um Ähnliches in einer On-Premises-Private-Cloud erreichen zu können, müssen ebenfalls gewisse Serverressourcen vorgehalten werden, um im Bedarfsfall schnell skalieren zu können.

Das bringt sowohl technische als auch organisatorische Hürden mit sich. Auch hier muss sehr viel auf Automatisierung gesetzt werden, um eine schnelle und einfache Skalierung der Systeme zu ermöglichen. Doch neben diesen Aspekten, die sich technisch gut lösen lassen, stellen sich auch die organisatorischen Herausforderungen, wie das Vorhalten der nötigen Software-Lizenzen. Das betrifft durchgehend sämtliche Komponenten wie Hypervisor und Betriebssystem bis hin zu einer Hadoop-Distribution. Gerade dieser Aspekt dürfte in vielen Unternehmen deutlich schwerer umzusetzen sein, da mehrere Abteilungen und Cost-Center involviert sind.

Doch natürlich gibt es diverse Gründe, warum es nicht ohne Weiteres möglich oder sinnvoll ist, eine Public Cloud einzusetzen. Eventuell sind die Datenschutzanforderungen des Unternehmens sehr individuell beziehungsweise extrem hoch, sodass die

FABIAN HARDT arbeitet als Senior Consultant bei der OPITZ CONSULTING Deutschland GmbH. Er hat langjährige Projekterfahrung in BI- und Analytics-Projekten und beschäftigt sich mit modernen Architekturen für die gestiegenen Anforderungen im Zeitalter der Digitalisierung.
E-Mail: fabian.hardt@opitz-consulting.com

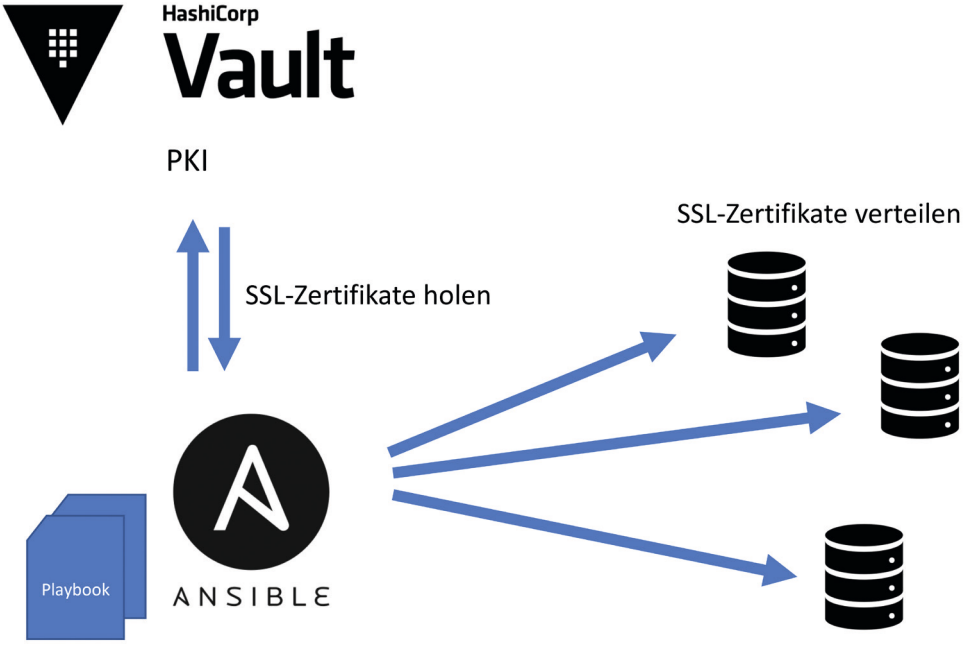

Abb. 1: Verteilung von SSL-Zertifikaten per Ansible

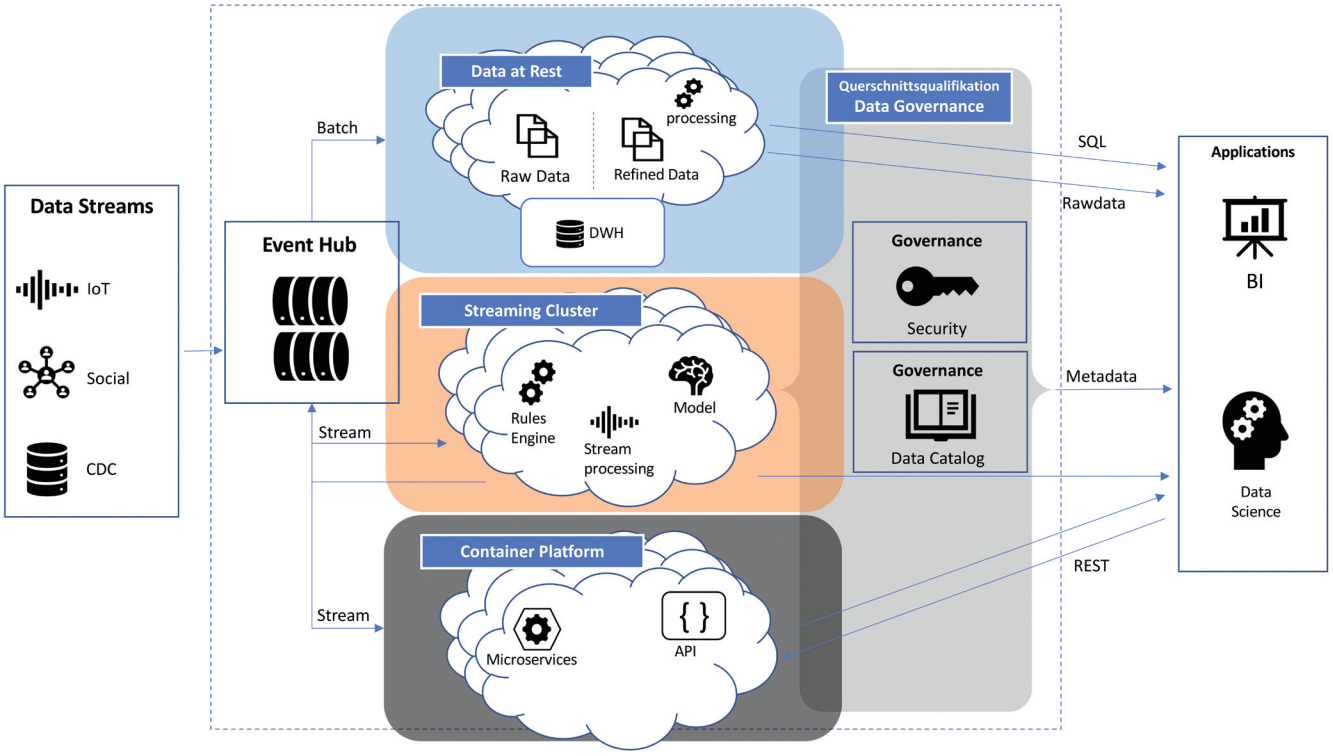

Abb. 2: Komponentenüberblick Analytics-Plattform

Entscheidung getroffen wird, die Plattform gänzlich offline im eigenen Rechenzentrum zu betreiben. In dem Fall kommt eine Public Cloud definitiv nicht in Frage. Ein weiterer Grund kann aber auch die restliche Systemlandschaft sein. Sind sämtliche Quell- und Zielsysteme, die an die Plattform angebunden werden sollen, On-Premises stationiert, wird eine Cloud-Plattform in der Regel zu hohen Kosten führen, da zumindest der ausgehende Datenverkehr bei den meisten Anbietern zu bezahlen ist. Insbesondere wenn das System eine gleichbleibende kontinuierliche Grundlast aufweist und daher nicht von Autoscaling Features der Public Cloud profitieren kann, kann eine On-Premises-Lösung weiterhin die günstigere Alternative sein [Ped20].

Bestandteile einer modernen Analytics-Plattform

Im Folgenden wird näher auf die Architekturbestandteile und Komponenten einer ganzheitlichen Analytics-Plattform-Lösung im eigenen Rechenzentrum eingegangen. Dabei wird eine praxiserprobte Private-Cloud-Architektur vorgestellt, die wir in einem Projekt umgesetzt haben: eine moderne Analytics-Plattform, die – um bei dem Eingangsszenario zu bleiben – Manuel das Leben sicher erleichtern würde, da der Aufbau sowie die wesentlichen Kernaufgaben des Betriebs automatisiert wurden. Gleichzeitig wird dargestellt, wer die Freunde des gelben Elefanten (Hadoop) sind.

Zunächst soll jedoch die Frage geklärt werden, ob eine Hadoop-Distribution überhaupt noch ausreichend ist, um die Anforderungen an eine moderne Analytics-Plattform abzudecken. Während alle gängigen Hadoop-Distributionen heute auch Streaming-Werkzeuge wie etwa Kafka mitbringen und auch mit Spark und Spark-Streaming schon eine Menge abdecken können, bleiben insbesondere im Bereich Datenintegration sowie Datenbereitstellung noch einige Lücken. Distributoren wie Cloudera reagieren darauf bereits mit Tools wie Apache NiFi, um die streambasierte Datenintegration aus diversen Quellen zu ermöglichen. Doch gerade bei der klassischen Datenintegration von relationalen Daten, beispielsweise aus relationalen Datenbanksystemen, tun sich große Lücken auf. Diese lassen sich mit der Auswahl eines gängigen ETL-Werkzeugs schließen oder durch individuelle Schnittstellen-Implementierung.

Doch der deutlich wichtigere Aspekt dürfte die Datenbereitstellung sein. Für viele Anwendungsfälle hat es sich bewährt, die Daten mit Hilfe sogenannter sogenannte Data APIs erreichbar zu machen. Diese können mit Hilfe eines API Gateways nach außen zur Verfügung gestellt werden und sorgen für einen standardisierten und gut abgesicherten Zugriffsweg auf die Daten innerhalb der Plattform. Sie bilden also eine Zwischenschicht zwischen den nachgelagerten Applikationen oder Data Consumption Tools und Self-Service-Werkzeugen wie Power BI oder Tableau. Sämtliche Security-Aspekte wie Authentifizierung und Autorisierung, also auch die benutzerdefinierte Einschränkung der Datensichtbarkeit, wird in diesem API-Layer abgehandelt. Außerdem stellt die entsprechende API-Spezifikation eine Art Contract zu den konsumierenden Zielsystemen dar.

Doch auch zur Bereitstellung von APIs, die gemäß modernen Entwicklungsparadigmen containerisiert entwickelt werden, bietet eine Hadoop-Distribution keine geeignete Ausführungseinheit. Somit wird eine Container-Plattform wie Kubernetes oder OpenShift für viele Anwendungsfälle unerlässlich. Mit einer solchen Plattform können

Softwarekomponenten auf einfache Weise hochperformant zur Verfügung gestellt und sogar ein Autoscaling implementiert werden.

Vor allem wenn die Analytics-Plattform diverse Quellsysteme hat, aus denen die Plattform die Daten bezieht, sollte ein sauberes Metadatenmanagement gewährleistet sein. Das ist sowohl für das spätere Auffinden der Daten in der Plattform wichtig als auch für die korrekte Umsetzbarkeit eines Rollen- und Berechtigungskonzepts, um den Datenzugriff entsprechend beschränken zu können. Hierzu kann entweder eine fertige Data-Catalog-Software zum Einsatz kommen oder ein Suchindex wie Elasticsearch als eigener Metadaten-Store genutzt werden. Diese Lösung bietet sich insbesondere dann an, wenn die enthaltenen Meta- und Event-Daten komplexer werden oder einem fachlichen Datenmodell zugrunde liegen. Elasticsearch kann hierbei ebenso für Nutzdaten zum Einsatz kommen, beispielsweise für eine große Anzahl an JSON-Dokumenten, die schnell und effektiv durchsuchbar sein sollen. Für große Systeme kann eine Kombination sinnvoll sein. Fachlich orientierte Event- oder Metadaten werden in einem Suchindex gespeichert und ein Data Catalog bildet eine ganzheitliche Sicht sowie Suchmöglichkeit auf die Metadaten der Plattform ab.

Eine weitere Querschnittsqualifikation ist ein Data Governance Layer, der zentral für den Zugriff auf die Daten zuständig ist. Dies wird notwendig, da Berechtigungen über verschiedene Systemgrenzen hinweg gültig sein und dort auch entsprechend umgesetzt werden müssen [Sor21]. Ein Governance Layer bildet also unter anderem eine Art Policy Generator ab, der Berechtigungs-Policies für verschiedene Komponenten generiert und verteilt. Sinnvollerweise ist dieser Layer eng mit dem Data Catalog verknüpft, sodass Metadaten genutzt werden können, um auf fachlicher Basis Berechtigungs-Policies zu generieren.

Eine weitere Aufgabe dieses Layers könnte auch die rechtzeitige Löschung von Daten sein – er unterstützt also ebenfalls bei der Umsetzung der geltenden Datenschutzanforderungen. Nicht zuletzt kann der Data Covernance Layer mit der Kombination aus Data Catalog und fachlichen Metadaten erheblich zur Bewertung der Data Quality beitragen. Ein Data Quality Score kann dem Benutzer im Data Catalog ausgewiesen werden und somit maßgeblich die Qualität der daraus resultierenden Reports erhöhen. Der Data Governance Layer ist hierbei eine Querschnittsqualifikation, die sich über alle Komponenten der Plattform hinweg erstrecken sollte. Das Schaubild in Abbildung 2 fasst die Komponenten einer Analytics-Plattform in einer Übersicht zusammen.

Es ist zu erkennen, dass verschiedene Dateneingänge in unterschiedlichen Geschwindigkeiten verarbeitet werden können. Im oberen Bereich findet klassisches Batch-Processing statt, wie man es auch aus einem Data Warehouse kennt. Im mittleren Teil ist ein moderneres Stream- und Event-Processing zu sehen. Mit Modellen kann in Echtzeit auf eingehende Daten reagiert und Folgeaktionen ausgelöst werden. Der untere Bereich stellt eine Containerplattform dar, die zur Ausführung von Microservices oder Data-APIs dienen kann. Zudem ist die Nutzung von BI-Tools und Analysewerkzeugen denkbar.

Praxisbeispiel

Im Folgenden wird eine konkrete Ausprägung der beschriebenen Architektur vorgestellt, die im Rahmen eines Projekts entworfen und umgesetzt wurde, mit dem Ziel, eine große Datenmenge in Echtzeit zu speichern und anschließend weiterzuverarbeiten (siehe Abbildung 3). Zum Großteil han-

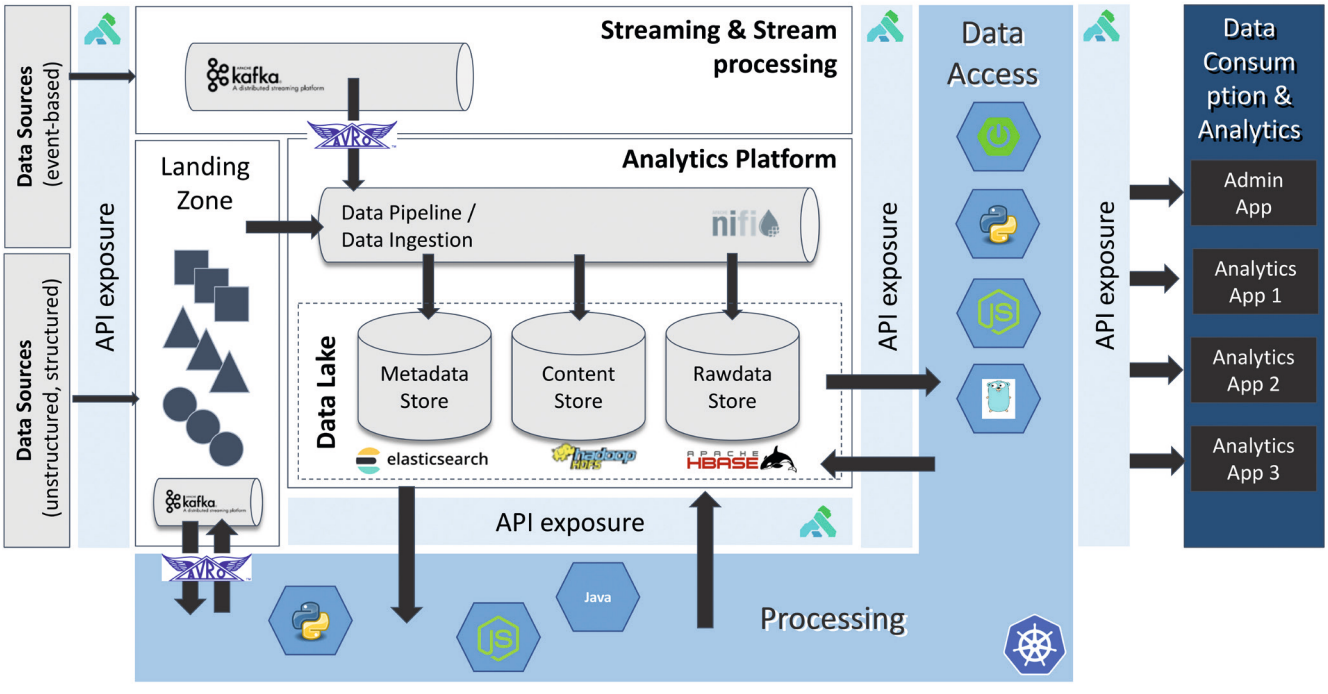

Abb. 3: Praxisbeispiel

delt es sich bei den Daten um semistrukturierte bis unstrukturierte Daten, die nach der Speicherung ein Preprocessing durchlaufen. In diesem Schritt werden automatisiert Metadaten erfasst sowie die Daten in Abhängigkeit zu ihrem Eingangskanal mit einem fachlichen Label versehen. Anhand dieses Labels werden automatisch Berechtigungen – gemäß den fachlichen Regelwerken – an entsprechende Benutzerkreise vergeben.

Als Hadoop-Distribution kommt ein Cloudera Stack zum Einsatz, HDFS und HBase werden zur Speicherung der Raw und Refined Data genutzt. Der fachlich genutzte Metadata Store wird mittels Elasticsearch abgebildet. Hier werden aus den Originaldaten extrahierte oder abgeleitete Metadaten indiziert und durchsuchbar gemacht. Der Data Ingest der unstrukturierten Daten erfolgt mittels Apache NiFi, das die Daten in entsprechenden Landing Zones abholt, entsprechende Metadaten erfasst und diese im Metadata bzw. Rawdata Store speichert. Jede neu eingeleitete Datei erzeugt ein Event im Eventhub, der mit Kafka abgebildet wird. Auf diese Events reagieren diverse Prozessoren und Machine-Learning-Modelle asynchron.

Jeder Schritt in dieser Verarbeitungskette wird in den Metadaten erfasst und kann in einem Data-Lineage-Graphen dargestellt werden. Somit ist eine lückenlose Darstellung des Datenflusses und der zugehörigen Transformationsschritte von Raw Data bis hin zu den Refined Data möglich.

Data APIs für den Zugriff auf die Plattforminhalte werden als Spring-Boot-Microservices entwickelt und in der Containerplattform OpenShift zur Ausführung gebracht. Um diese entsprechend abzusichern und nach außen freizugeben, kommt das Kong API-Gateway zum Einsatz. Ein entsprechendes Rollen- und Rechtekonzept wurde ebenfalls mittels API abgebildet. Diese generiert und verteilt entsprechende Berechtigungs-Policies über die Systemgrenzen der Einzelkomponenten hinweg. So kann der Zugriff einheitlich sowohl auf Nutzdaten in HBase und HDFS begrenzt werden als auch auf Metadaten in Elasticsearch.

Anhand dieses Beispiels wird schnell deutlich, wie komplex eine solche ganzheitliche Analytics-Plattform werden kann. Allein durch die Anzahl verschiedener Komponenten, die in sich zumeist clusterbasierte Systeme sind, wird der Betrieb zu einer echten Herausforderung für Manuel und seine Kollegen. Es ist also nicht ausreichend, allein die Installation der Komponenten zu automatisieren, es sollten ebenfalls die gängigsten Administrationsaufgaben in Form von Skripten abgebildet werden, um die Administratoren bestmöglich zu entlasten. Schon die regelmäßigen Updates der Plattformkomponenten werden Manuel zu einem Großteil seiner Arbeitszeit auslasten.

Nicht zuletzt ist aber auch ein möglichst lückenloses Monitoring wichtig, um Fehlerfälle schnell erkennen zu können. Im Idealfall wird dieses Monitoring um eine zentrale Logserverkomponente erweitert, sodass die wichtigsten Logfiles der diversen Systeme zentral gesammelt und ausgewertet werden können. Das bietet bei der Fehlersuche auf dieser skalierbaren Infrastruktur erhebliche Geschwindigkeitsvorteile. Dies muss sich jedoch nicht auf Infrastrukturkomponenten begrenzen, es können auch Metriken und Logs der eigenen API-Services angebunden und ausgewertet werden. In Kombination mit Data Lineage lassen sich auch fachliche Verarbeitungsfehler schnell erkennen und eingrenzen.

Fazit

Zusammenfassend lässt sich sagen, dass sich der Funktionsumfang einer Public Cloud auch On-Premises erreichen lässt, auch wenn hier einige Hürden mehr zu nehmen sind, beispielsweise beim Thema Reserve-Hardware sowie der schnellen Skalierung von Lizenzen. Im Architekturüberblick wurde eine Auswahl an möglichen Komponenten vorgestellt, mit denen sich eine moderne Analytics-Plattform umsetzen lässt. Maßgeblich für den stabilen Betrieb ist allerdings ein hoher Grad an Standardisierung sowie Automatisierung. Wiederkehrende Wartungs- und Betriebsaufgaben sollten das Administrationsteam nicht belasten, damit ausreichend Zeit für regelmäßige Security-Patches und Software-Updates der vielen verschiedenen Komponenten bleibt. Außerdem ist ein stabiler Betrieb der unteren Level, wie Hardware, Hypervisor und Betriebssystem, sehr wichtig und nimmt bereits einen festen Anteil an Ressourcen ein. Fast bedeutender jedoch ist ein möglichst lückenloses Monitoring sämtlicher Plattformkomponenten, um Fehlerfälle schnell und zuverlässig erkennen zu können.

Der gelbe Elefant (Hadoop) und seine Freunde (Kubernetes, Elasticserach, API-Gateways, Consumption-Tools etc.) lassen sich also auch On-Premises – in einer Private-Cloud-Lösung – durchaus noch beherrschen, allerdings mit den genannten Hürden und einem initialen Mehraufwand beim Aufbau der Plattform.

Quellen

[Ped20] Pedamkar, P.: Cloud Computing vs Hadoop. 2020, https://www.educba.com/cloud-computing-vs-hadoop/, abgerufen am 11.9.2021

[Sha17] Shaik, S.: Hadoop On Cloud vs Hadoop On Premises. 21.1.2017, https://www.linkedin.com/pulse/hadoop-cloud-vs-premises-shahebaz-shaik, abgerufen am 11.9.2021

[Sor21] Sorge, L.: A Comprehensive Guide to Governed Data Lakes. 3.5.2021, https://www.codemotion.com/magazine/dev-hub/data-scientist/governed-data-lakes-guide/, abgerufen am 11.9.2021

[Ste20] Stender, D.: Cloud-Infrastrukturen: Infrastructure as a Service – So geht moderne IT-Infrastruktur. Rheinwerk Computing 2020