



Pipeline zur Stock-Price Analyse durch Machine Learning und Big Data Integration

Tim Willkens

IU14073577

30-Jun-24

Temmenhausen

Inhaltsverzeichnis

INHALTSVERZEICHNISS	2
1. MOTIVATION.....	3
2. SYSTEMARCHITEKTUR	3
2.1 DATEN-QUELLE	4
2.2 MICROSERVICES	4
2.3 CI/CD	5
LITERATURE.....	6

1. Motivation

Die technische Chart-Analyse ist ein wesentlicher Bestandteil der Finanzanalyse und wird häufig von Händlern und Investoren verwendet, um Trends zu identifizieren und Investitionsentscheidungen basierend auf der technischen Analyse zu treffen. Allerdings kann diese Analyse zeitaufwendig und komplex sein, insbesondere wenn sie auf einer großen Menge von Daten und verschiedenen Stocks basiert. Hier kommt die Integration von Machine Learning und Big Data ins Spiel.

Die Integration von künstlicher Intelligenz, Machine Learning Algorithmen oder Deep Learning Funktionen hebt die Datenanalyse auf ein neues Level, wenn es darum geht, Daten aus unterschiedlichen Quellen miteinander zu kombinieren, um daraus neue Informationen zu gewinnen, Trends zu erkennen oder Prognosen abzuleiten.

Ziel des Portfolios soll sein, eine Daten-Pipeline auszubauen, welche Daten mehrerer Aktien in nahezu Echtzeit bereitstellt und diese an ein Machine-Learning Modell sowie an eine Visualisierungs-Plattform/ interaktives Dashboard weiterleitet. Das Machine-Learning Modell und das Dashboard sind nicht Teil des Portfolios.

2. Systemarchitektur

Die Implementierung der Datenpipeline soll nach gängigen DevOps Prinzipien erstellt werden. Dies ermöglicht das Projekt robust, verlässlich, skalierbar und wartbar herzustellen. Hierzu werden moderne Methoden wie Microservices (MS), Infrastruktur as Code (IaC) sowie CI/CD-Pipelines verwendet. Die Grundsätzliche System-Architektur ist in Abbildung 1: System-Architektur zu sehen.

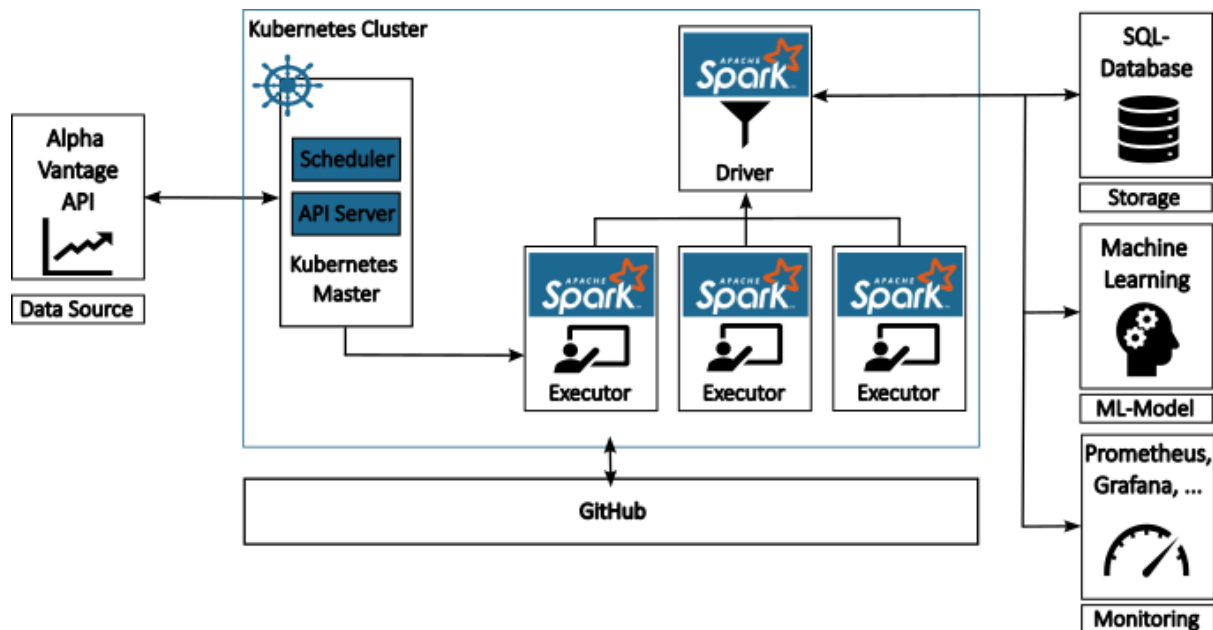


Abbildung 1: System-Architektur [7]

2.1 Daten-Quelle

Die Daten für das Projekt werden über die Open-Source API **Alpha Vantage** [1] bezogen. Die Alpha Vantage API ist eine cloudbasierte Schnittstelle, die Echtzeit- und historische Finanzmarktdaten bereitstellt sowie technische Indikatoren abzurufen, was sie zu einem wertvollen Werkzeug für Finanzanalysen und -prognosen macht [1].

Die Daten von Alpha Vantage werden mit Open-Source-Framework **Apache Spark** [2] verarbeitet. Apache Spark bietet eine schnelle und flexible Möglichkeit, Daten in Echtzeit zu analysieren und zu verarbeiten. Weitere Merkmale von Spark sind:

- Analyse-Engine
- Unterstützung mehrerer Programmiersprachen
- Schnelle Verarbeitung
- Skalierbarkeit und Flexibilität
- Resilient Distributed Datasets
- Vielzahl an Bibliotheken

2.2 Microservices

Microservices sind ein moderner Ansatz in der Softwareentwicklung, bei dem Anwendungen aus einer Sammlung kleiner, unabhängiger Dienste bestehen, die über klar definierte APIs

kommunizieren. Sie ermöglichen eine flexible Skalierbarkeit, da jeder Service unabhängig voneinander skaliert werden kann, um die Nachfrage zu decken [3].

Docker und **Kubernetes** sind zwei zentrale Technologien, welche die Implementierung und Verwaltung von Microservices erheblich erleichtern.

Docker ist eine Open-Source-Plattform, die es Entwicklern ermöglicht, Anwendungen in Containern zu erstellen, zu deployen und auszuführen [4]. Container sind hierbei leichtgewichtige, portable Umgebungen, die alles enthalten, was eine Anwendung zum Laufen braucht, einschließlich Code, Bibliotheken und Abhängigkeiten.

Kubernetes ist eine Open-Source-Plattform zur Orchestrierung von Container-Anwendungen. Es automatisiert die Bereitstellung, Skalierung und Verwaltung von containerisierten Anwendungen über ein Cluster von Knoten hinweg [5].

2.3 CI/CD

Versionierung ist der Prozess, bei dem Änderungen an Dateien und Projekten verfolgt werden und somit ein wichtiges Werkzeug in der Software-Entwicklung. Git ermöglicht es Entwicklern, die Änderungshistorie zu verfolgen, zu sehen, wer welche Änderungen vorgenommen hat, und bei Bedarf zu früheren Versionen zurückzukehren [6].

Weiter unterstützt GitHub CI/CD-Pipelines. Es automatisiert den Prozess des Buildens, Testens und Bereitstellens von Codeänderungen, um diese schneller und zuverlässiger zu veröffentlichen.

Literature

- [1]: <https://www.alphavantage.co> (last access 2024-06-30)
 - [2]: <https://spark.apache.org/> (last access 2024-06-30)
 - [3]: <https://aws.amazon.com/de/microservices/> (last access 2024-06-30)
 - [4]: <https://www.docker.com/> (last access 2024-06-30)
 - [5]: <https://kubernetes.io/> (last access 2024-06-30)
 - [6]: <https://docs.github.com/en/get-started/using-git/about-git> (last access 2024-06-30)
 - [7]: <https://aws.amazon.com/de/blogs/compute/running-cost-optimized-spark-workloads-on-kubernetes-using-ec2-spot-instances/> (last access 2024-06-30)
-