# Geo IP Locator

James Pedersen - 7790128

## 1.1 Technologies

Bei diesem ETL Pipeline wurden einige Technologien zusammenverwendet, um eine funktionstüchtige Anwendung zu produzieren. Im wesentlichen ist Docker Klebstoff, der die Virtualisierung von Anwendungscontainer ermöglicht. Der ETL Prozess wird vom Airflow in Docker gesteuert. Über Zugriff auf diesen container (localhost:8080) kann der ETL Prozess Importdata ausgeführt werden.

Die Daten werden über Hive in HDFS (ebenfalls containerisiert) zwischen gespeichtert und bearbeitet, bevor sie in die MYSQL Datenbank in endgültigen Tabellen gespeichert werden. Über diese Datenbank werden die Daten vom Python Flask Backend gequeried. Der Javascript Frontend greift auf die Backend Rest Endpoints bei Flask für die Location und IP Funktionalitäten.

Es gibt dementsprechend 3 Dockercontainer. Ein Container für Airflow, Hadoop und MySQL. Flask Backend ist ein vergleichbar klein / Lightweight Rest API. Die Abhängigkeiten dessen sind in einer Requirementsdatei aufgelistet. Zusätzlich ist der Frontend vergleichbar lightweight und kann über einen Browser verwendet werden. Es ist vorgesehen, dass die Container an der gleichen Machine laufen. Dies kann lokal oder in der Google Cloud ausgeführt werden.

Die nötigen Images sind gebaut und die Einstellungen der Container in einer docker-compose Datei aufgelistet. Dementsprechend können alle ETL Container mit einem Befehl hochgefahrend werden.

## 1.2 Installation

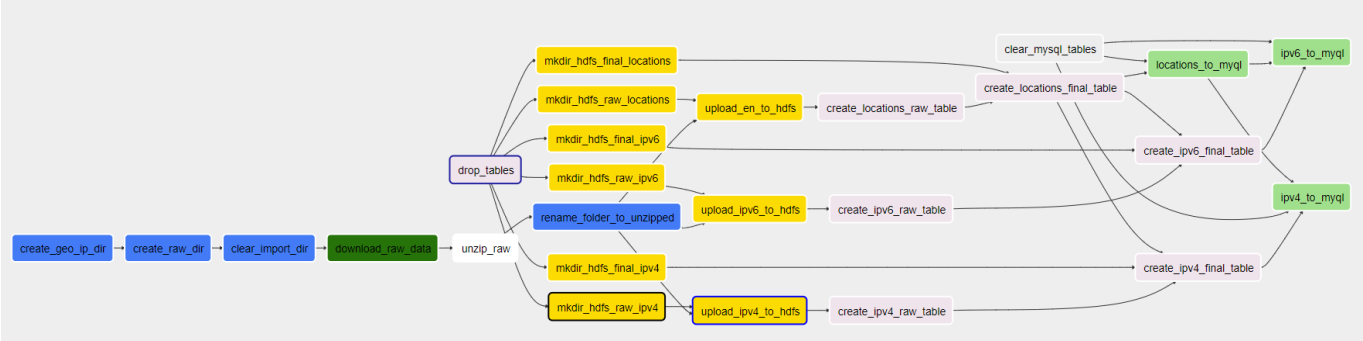
Die nötigen Dateien, um die EtL Pipeline und Anwendung auszuführen sind in der angehängten Zipdatei. Diese muss entzippt werden. Um die ETL Container hochzufahren muss am Rechner Virtualisierung aktiviert und Docker installiert sein. Im Hauptordner kann dann der Befehl „docker-compose up -d“ ausgeführt werden. Anschleißend (nach dem erfolgreichen Hochfahren des Hadoop Containers) soll eine Bashkonsole im Hadoopcontainer mit „docker exec -it hadoop bash“geöffnet werden. Somit können die Hadoop und Hive services gestartet werden. Dafür werden „sudo su hadoop“, „start-all.sh“, „hiveserver2“ ausgeführt werden. Wenn diese Befehle in der Reihenfolge erfolgreich durchgelaufen sind, kann der ETL Prozess ausgeführt werden.

Bei der Geo-IP Anwendung muss python installiert sein. Dazu müssen die Abhängigkeiten installiert werden. Um das zu machen, kann im Hauptorder der Befehl „pip install --no-cache-dir -r .\docker\ipservice\src\requirements.txt“.

## 1.3 Ausführen

Der DataImport Dag kann über „localhost:8080/admin“ ausgeführt werden. Nachdem die Pipeline durchgelaufen ist, kann die Anwendung verwendet werden. Um den Flask Backend zu starten, kann der Befehl „Python .\docker\ipservice\src\rmain.py“ ausgeführt werden. Die HTML Seite kann dann geöffnet werden, die sich im site Ornder befindet. Die Anwendung kann eine Location für beliebeige Ipv4 und ipv6 Adressen finden. Zuerset wird die eigene angezeigt (funktioniert bei einer lokalen Verbindung nicht). Über ein Textfeld und Suchbutton können die Adressen belibiger Ipaddressen abgefragt werden.

## 1.4 ETL Pipeline




Bei dem ETL Pipeline wurden viele Custom Operatoren verwendet. Der Pipeline folgt dem Fluss: Downloadordner erstellen, downloaden, entzippen, umbenennen, hive ordner Erstellen, hive Tabellenfüllen, Tabelleninhalt Transformieren in endgültige Form, Daten von Hive auf MySQL umziehen. Dazwischen gibt es jedoch ein paar zusätzliche Abhängigkeiten. Beispielsweise sind die Ipv4 und ipv6 Tabellen von Location abhängig, weil die Fremdschlüssel in den IP MySQL Tabellen mit den Primary Keys aus der Locationstabelle übereinstimmen müssen, um den Prozess erfolgreich auszuführen. Deshalb müssen ein paar Transformationen druchgeführt werden, die , diese Abhängigkeiten von Locations einführen.

Ein paar Custom Operatoren mussten geschrieben werden, um diese Pipeline zu realisieren. Eins davon is der Rename Operator, da der name des entzippten Orders nicht bekannt ist. Deshalb musste dieser über ein Pattern erkannt werden und in einen bekannten Namen umbennant werden.

Zusätzlich musste der Unzip Operator umgeändert werden, denn nur die erste Schicht von Dateien mit dem Stanard Unzip Operator entzippt werden. Um das rekursive  Entzippen zu ermöglicen mussten di Daten aller Unterdateien rekursiv erkannt werden.