Deploying Data Pipelines using Kubernetes on Azure

Yuvraj Gupta[1], Roshan Lalwani[2], Rajat Mittal[3], Karma Patel[4], Daniel Whitenack[5]

Purdue University Krannert School of Management

yuvraj@purdue.edu[1]; rlalwan@purdue.edu[2]; mittal51@purdue.edu[3]; karma@purdue.edu[4]; dwhitena@purdue.edu[5]

Abstract

The goal of this project is to build an ETL infrastructure on the MS Azure cloud platform for our client: A leading healthcare consulting firm that processes massive amount of healthcare data on a routine basis. Their current framework relies on data sources that are decentralized and lack robustness. We are facilitating the commoditization of this data in the cloud by creating Data Pipelines managed by Kubernetes and Pachyderm. Our proof of concept will give the client a centralized and robust analytics and BI platform with the following key features:

- **Centralized**
- **Fault tolerant**
- **Easy to implement and replicate**

Introduction

Building **scalable** and **easy to maintain** data pipelines is one of the most important steps for any analytical process. In our project, we are developing a **proof of concept** model for automating the process of files validation on a routine basis and feeding the validated data to a **centralized data hub**. Broadly speaking, this project can be broken down into three different **technical problems**:

- **New Data:** Automatically fetching new files as and when they enter Blob Storage
- **Spin Containers:** Using **Docker Containers** to build the software packages as a set of microservices which provides high scalability and eliminates a single point of failure in the pipeline
- **Data Validation:** Separating the data into one of the pre-defined business file types and applying validation steps by sending it through the pipeline
- **SQL Hub:** The centralized location where all the transformed data is stored for ease of use in terms of **Business Intelligence reporting** and providing **data platforms** as products to clients

Fig 1: The Flow of Development

Literature Review

We conducted thorough literature review on the following topics:

Containerized Data Pipelining Using Docker: Docker is a software platform that allows us to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime (Learn More: https://www.docker.com/why-docker)

Kubernetes Deployed on MS Azure: These are the tools we use to orchestrate our Docker containers and manage our compute and storage resources effectively. (Learn More: https://kubernetes.io/docs/home/)

Pachyderm: A thin layer built on top of Kubernetes that lets you deploy and manage multi-stage, language-agnostic data pipelines while maintaining complete reproducibility and provenance. (Learn More: https://pachyderm.readthedocs.io/)

Methodology

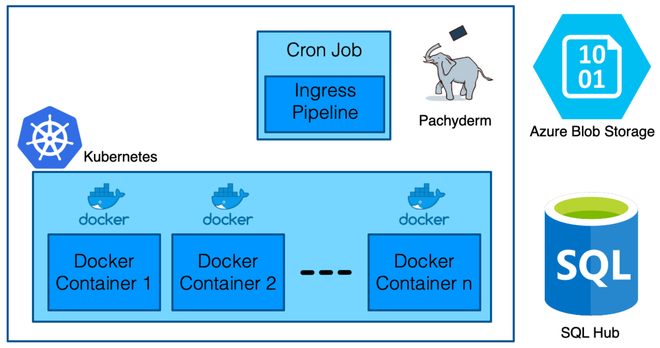

Fig 2: The Toolkit We Used

The broad **development steps** followed to build the architecture are as follows:

1) We created a **blob storage** space on MS Azure to store our flat files
2) We then spun up a **virtual machine** instance
3) We then created a **Docker image** which "listens" to our blob storage which is run periodically using **Cron**
4) When a new file enters blob storage, the script inside the docker image transfers the file to an "**Input Repository**"
5) Upon receiving a file in the Input Repository, **Pachyderm** will prompt **Kubernetes** to run the docker containers with our data validation pipeline
6) Once the pipeline is run, it pushes the metadata into an "**Output Repository**"
7) After going through this pipeline, the transformed data is then pushed into a **centralized SQL hub** that we have created

As cloud computing continues to gain massive mainstream popularity, customers demand the ability to have **hybrid computing**, which is to say they don't want to be locked in to any one cloud provider and want to be able to either use multiple platforms at once or move across platforms with ease.

That's why it was important for us to create a solution that is not only replicable on the same platform (MS Azure) but one that can be easily implemented **across different cloud platforms**. We tested an Azure specific architecture **(Fig 3.)** but finally implemented the architecture that is scalable across cloud platforms **(Fig 4.)**

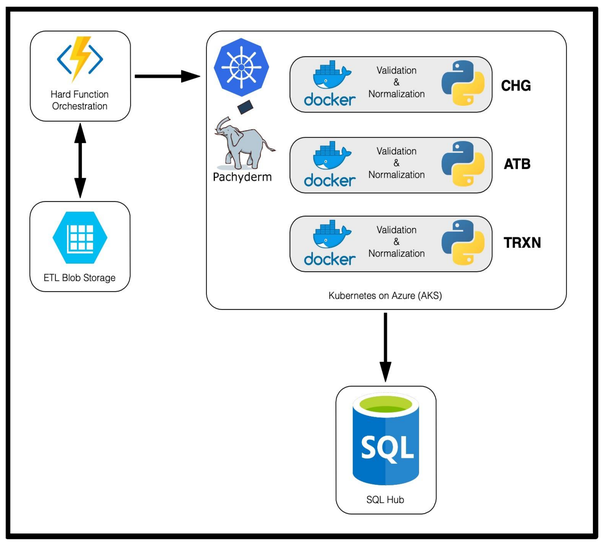

Fig 3: Azure Specific Skeletal Architecture

Results

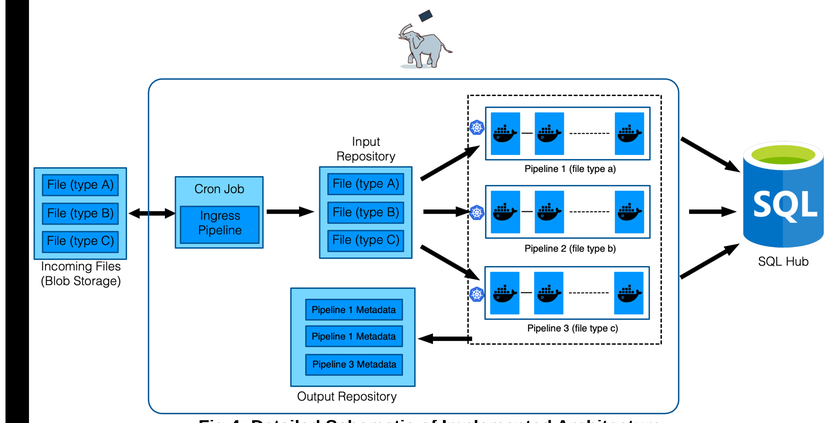

Fig 4: Detailed Schematic of Implemented Architecture

Salient Features Implemented:

- We built separate sample pipelines for the client's accounts receivable data (**Charge and Transaction Pipelines**)
- Each validation code was written in **Python** and **modularized** with **Docker**
- We also used Python to write an **Ingress Function** that always "listens" for new data to validate through the pipeline
- We fed the client's charge and transaction data through our validation pipeline to prove that it works seamlessly for transforming the data

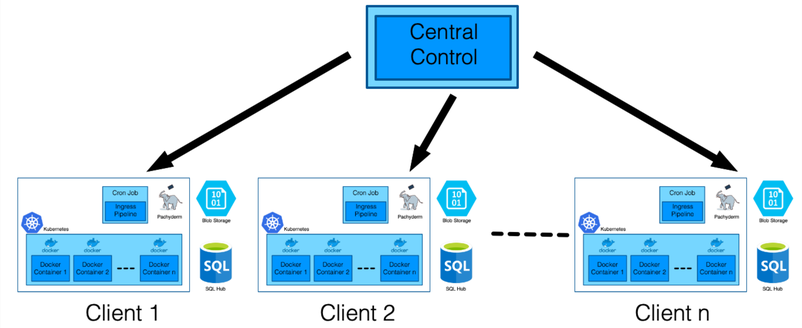

Fig 5: Reproducibility across platforms

Conclusions

Projected Impact:

- **Easy Scaling:** The resources can be scaled up or down effortlessly
- **Effective Computing:** Maximum efficiency in resources paid for due to Kubernetes orchestration
- **Stability & Fault Tolerance:** The infrastructure is robust and a breakdown in any one part will not cause the system to go down completely
- **Access Control:** The client can share and control access globally

Scope: Revenue Cycle Analytics **(RCA)** is our client's main product which will be hosted on this architecture. In the past, the data required for this product has been stored de-centrally. Our centralized architecture will enable near real time access to all this data to all hospitals via a web application. This presents immense value to hospitals (our client's customers) can use this data to accelerate monthly close processes, improve analyses, and monitor and measure revenue cycle activity seamlessly.

Acknowledgements

We thank Professor Daniel Whitenack for constant guidance on this project.