**P34 - OCR POC Tasks**

**Developer:** Vinay Gandra

**People Involved:** ### – Client POC Stakeholders, ### – HWX Admin

**Project:** ###

**POC:** OCR Processing on Sensitive Data in secure Hadoop datalake

**Development Area:** BI Lab Environment

**Problem:** Long processing of many image documents to text using traditional methods.

**POC Scope:** Develop Spark jobs to distribute processing of images in parallel across the cluster. Images and processed data to be secured with encryption.

**Data Type:** Sensitive, Critical

**Requirements:**

1. **Security –** Encrypt all Sensitive data and store data in restricted areas
2. **Data Ingestion –** From source server to datalake
3. **Data Transformation –** For the purpose of indexing, tracking and retrieval to match further data processing needs
4. **Data Processing –** Perform OCR on image data to extract text
5. **Data Access –** Make processed data easily available for further text processing requirements

**Tasks:**

**Security setup & test**

1. **HDFS Encryption:**

**Technology:** JAVA Cryptography Encryption(JCE) Unlimited Strength with AES upto 256 bit key length encryption

**Provider:** Hadoop Transparent Data Encryption

**Key Management:** Ranger KMS

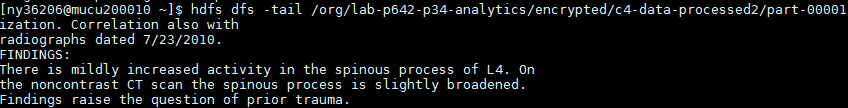
**Method:**

Create encryption zones with encryption keys managed by Ranger KMS for data to be provided restricted access. Limit key access to authorized user through IAM.

**Encryption Zone(EZ):** /org/lab-p642-p34-analytics/encrypted/

1. **Encryption Test:**

**Authorized Attempt:**



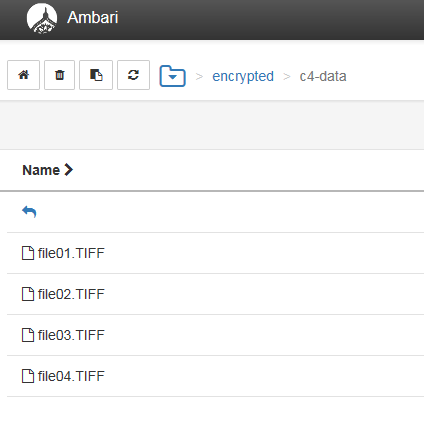
**Unauthorized Access Attempt:**



**Data Ingestion**

Manual ingestion of data from source to datalake through Ambari File Views

**Location:** /org/lab-p642-p34-analytics/encrypted/c4-data



**Proposal:**

Develop automated ingestion jobs from source ftp server to restricted areas in datalake using existing like Apache Nifi

NFS Mount windows server on edge nodes of cluster

**Data Transformation**

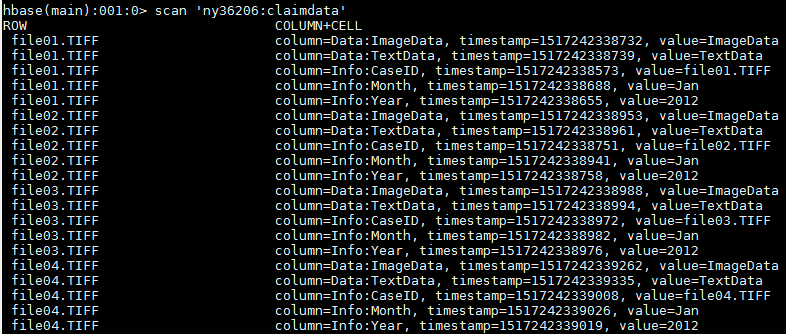
1. **HBase Metadata:**

Flatten source data directory hierarchy Year>Month>CaseID to HBase columnar format for better storage and access patterns.

**HBase Table Structure:**

**Name –** Hbase Namespace : TableName

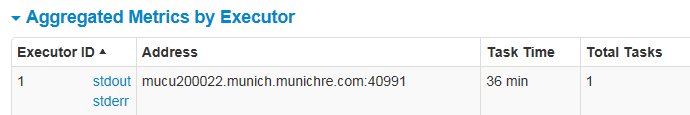
**Column Families/ Columns –** Info:CaseID, Info:Year, Info:Month, Data: ImageData, Data: TextData



**Data Processing**

1. **Configuration –** Build Tesseract Java Wrapper TESS4j for OCR processing. Configure Spark to load Tess4j libraries at run time
2. **Spark Job –** Develop Spark job to process image file in parallel
3. **Zeppelin –** Provide users with Zeppelin notebook with the Spark OCR code for adhoc queries and further development
4. **Performance**

**Single file(300 Pages) processing times – 36 Minutes**



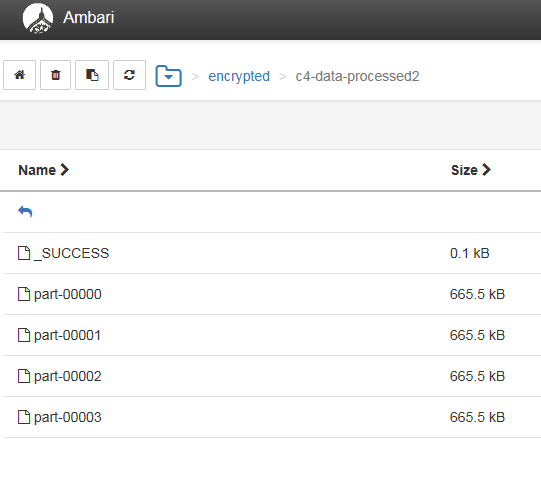
**Multiple files(300 Pages each) processing times – Average 40 Minutes**



1. **Limitations –** Limited by resources and scope limited to multifile parallel processing. Single file with many pages processing is not distributed for pages to be extracted in parallel.
2. **To do:** 
   1. Read file paths from HBase, store status, metadata and processed text data back to HBase and text data to Hive
   2. Split files into pages and store each page as column in HBase
   3. Parallel process both files and pages in files
   4. Develop Hive data retrieval patterns

**Data Access:**

**Current:** Processed text Stored as text files in encrypted / restricted zones in datalake



**Proposal:**

Store each page of processed text in the corresponding HBase table file reference

Combine all pages of file and save full text to Hive for further text processing for Machine learning tasks