IBM GEN AI REPORT

Phase 3-Final Report : AI-Powered Document Digitization and Analysis using IBM Watsonx

VIT Bhopal University

Name : Asmita Sarkar
Registration Number : 22BSA10100
Email : asmitasarkar2022@vitbhopal.ac.in
Github link :
https://github.com/Asarkar27/AI-Powered-Document-Digitization-and-Analysis-using-IBM-Watsonx

AI-Powered Document Digitization and Analysis using IBM Watsonx

---

Final Report: Intelligent Text Extraction from PDF Documents using IBM Watsonx.ai

---

Table of Contents:

---

1. Introduction

In the contemporary digital age, information is often trapped in unstructured formats such as PDF documents. While these files offer portability and consistent formatting, they pose significant challenges when it comes to data extraction, transformation, and analysis. This project titled **"Intelligent Text Extraction from PDF Documents using IBM Watsonx.ai"** addresses this issue by leveraging AI-driven capabilities to automate and optimize the extraction of text and tabular data from PDFs. The goal is to develop a robust, scalable, and accurate text extraction pipeline using IBM's powerful Watsonx.ai platform integrated with IBM Cloud Object Storage (COS).

This report documents the detailed execution of Phase 3 of the project, where the final product is tested, validated, and submitted for evaluation. The work showcases the seamless integration of Watsonx.ai with object storage and outlines the comprehensive process involved in setting up, deploying, and executing a complete text extraction job through an API-based Python implementation.

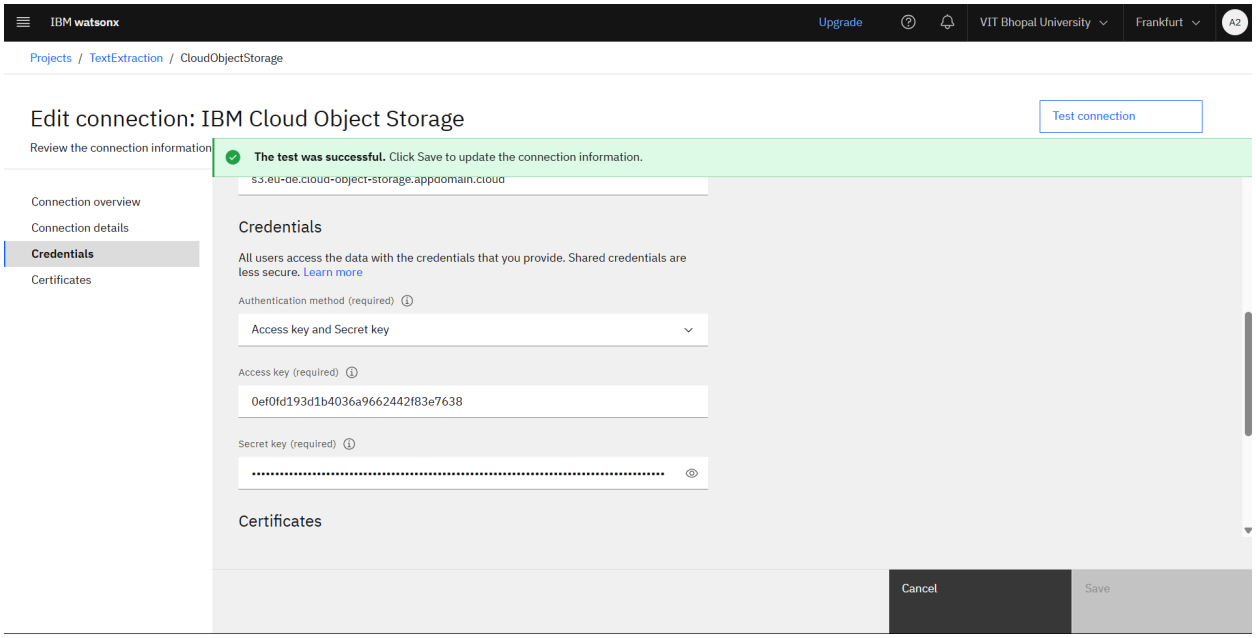
---

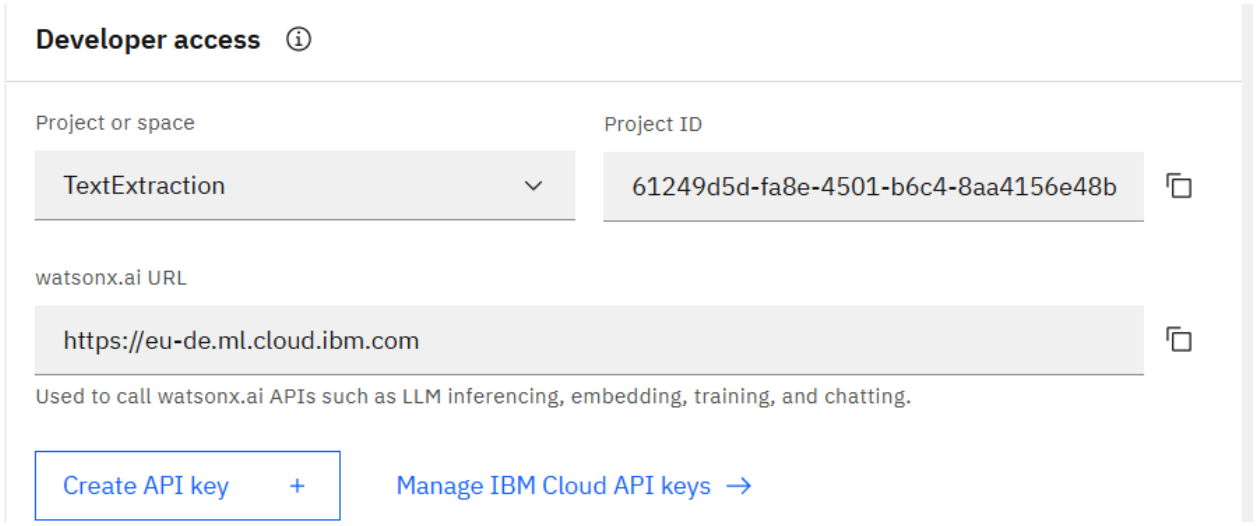
2. Objectives of the Project

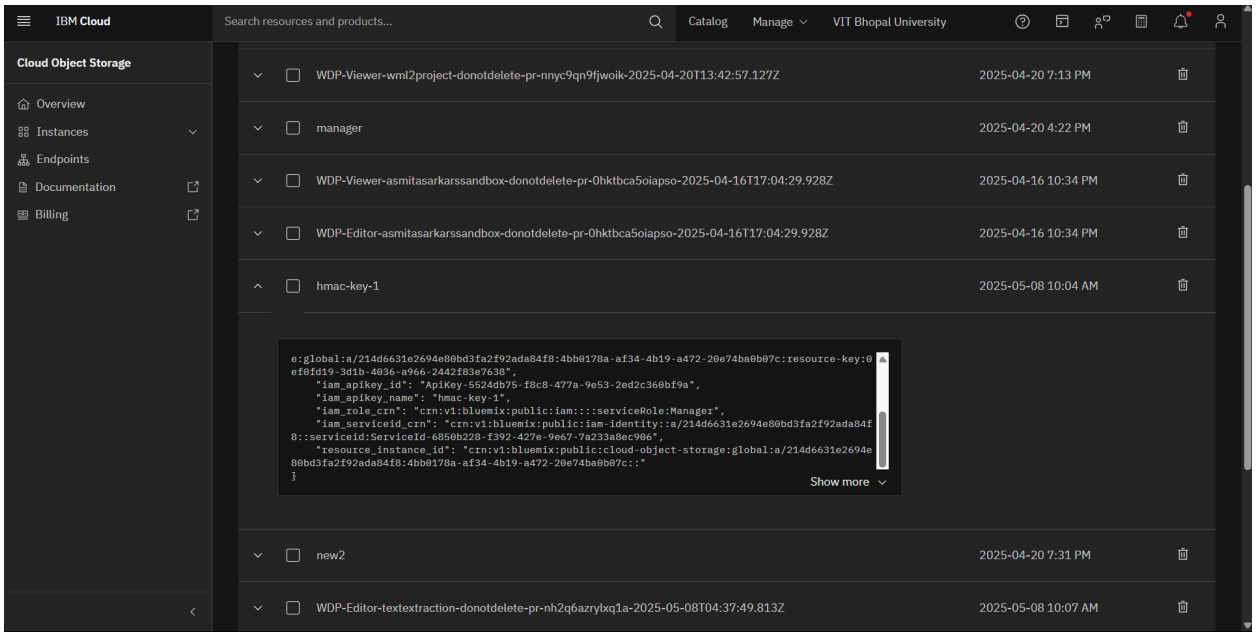
The core objectives of this project are:

- To establish a functional pipeline for extracting textual and tabular data from PDF files using IBM Watsonx.ai.
- To automate the data flow by uploading and downloading files to and from IBM Cloud Object Storage (COS).
- To leverage IBM's foundation models to improve the quality of OCR (Optical Character Recognition) and table extraction.
- To structure the output in a markdown format that retains document formatting for easy parsing and visualization.
- To ensure scalability and modularity in code, allowing extension to multiple document types in the future.
- To validate the output against original documents to ensure data accuracy.

This report focuses on the deliverables and learnings from **Phase 3**, which constitutes the final implementation, results, evaluation, and submission.


IBM watsonx
Upgrade
VIT Bhopal University
Frankfurt
A2
Projects / TextExtraction / CloudObjectStorage
Edit connection: IBM Cloud Object Storage
Test connection
Review the connection information
The test was successful. Click Save to update the connection information.
Connection overview
Connection details
Credentials
Certificates
s3.eu-de.cloud-object-storage.appdomain.cloud
Credentials
All users access the data with the credentials that you provide. Shared credentials are less secure. Learn more
Authentication method (required)
Access key and Secret key
Access key (required)
0ef0fd193d1b4036a9662442f83e7638
Secret key (required)
Certificates
Cancel
Save





---

3. Tools and Technologies Used

Tool/Technology	Purpose
IBM Cloud Object Storage	To store PDF documents and extracted results securely

IBM Watsonx.ai	To process text extraction through its foundation models
Python SDK (ibm-watsonx-ai)	For programmatic interaction with Watsonx.ai
Boto3	Interface with IBM COS via AWS S3-compatible API
Jupyter Notebooks / Python Scripts	For development, testing, and execution of code
Markdown	Format for representing extracted data
GitHub	Code version control and collaboration

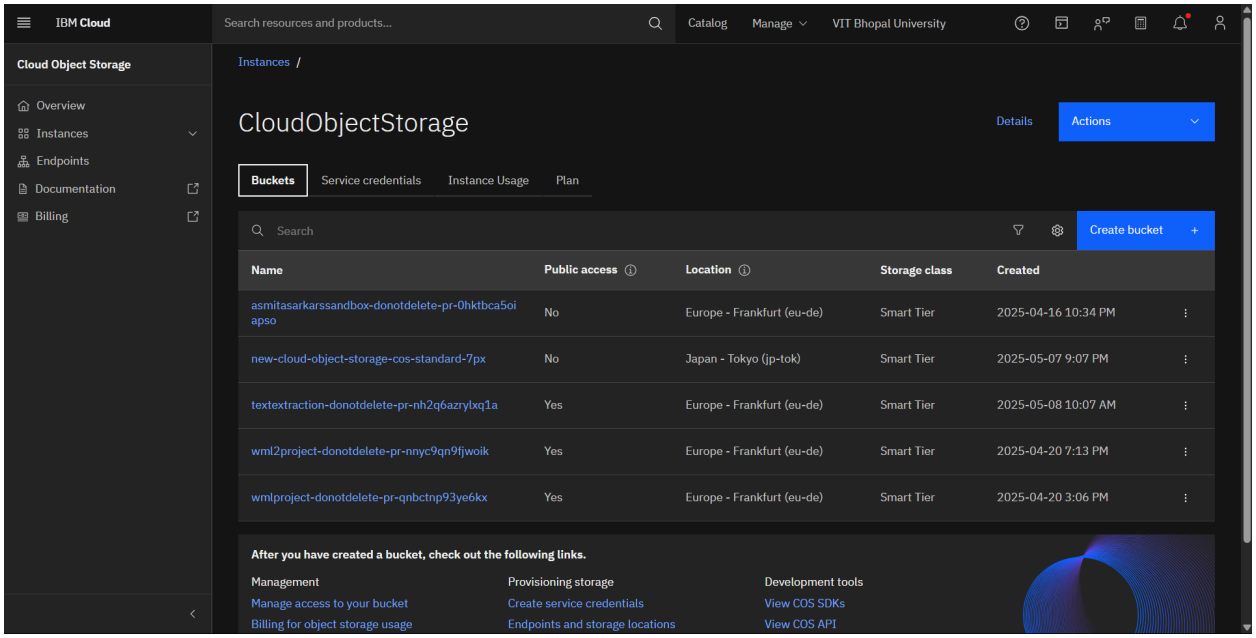
---

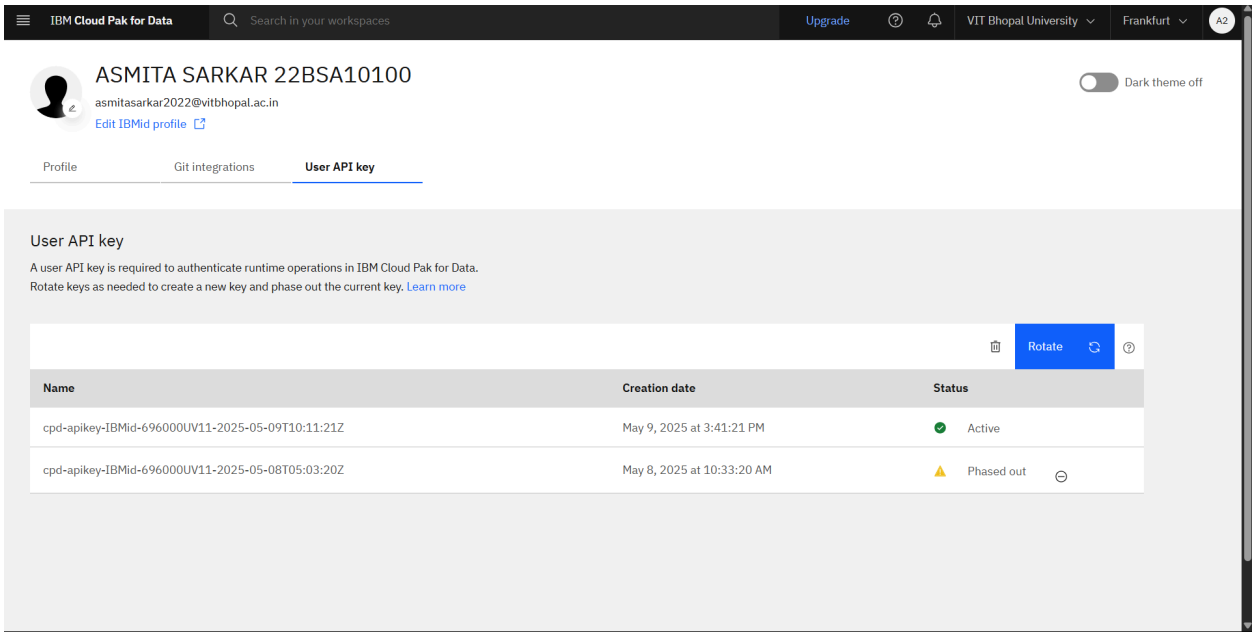
4. Project Planning and Phase Overview

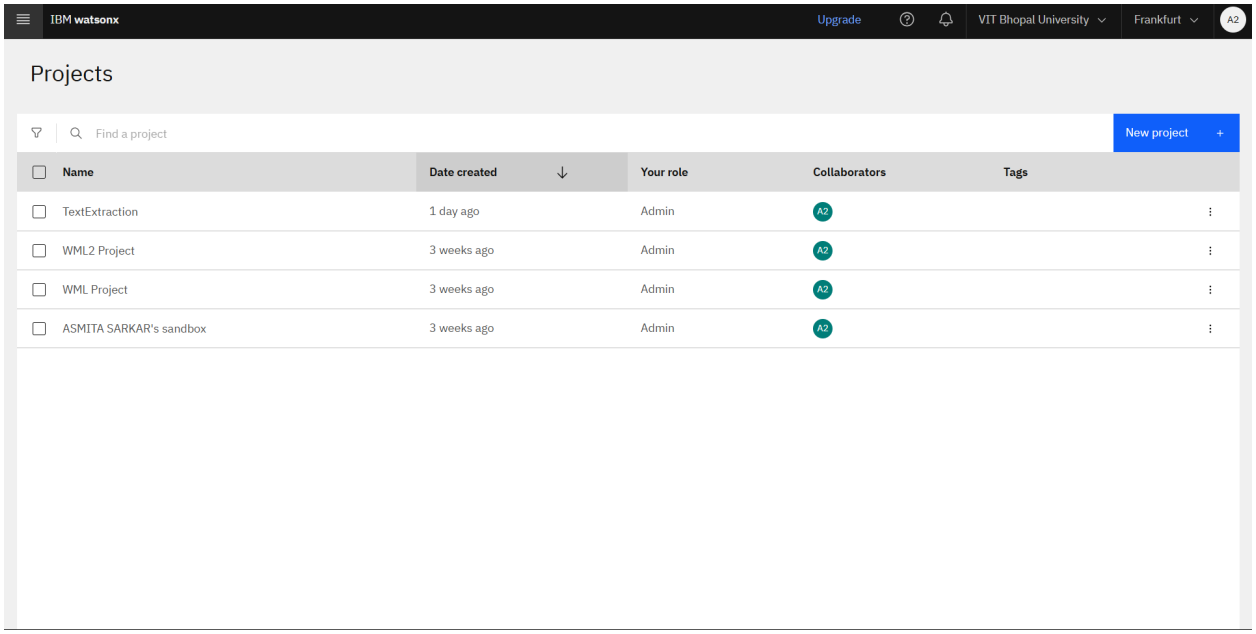
The project was divided into three main phases:

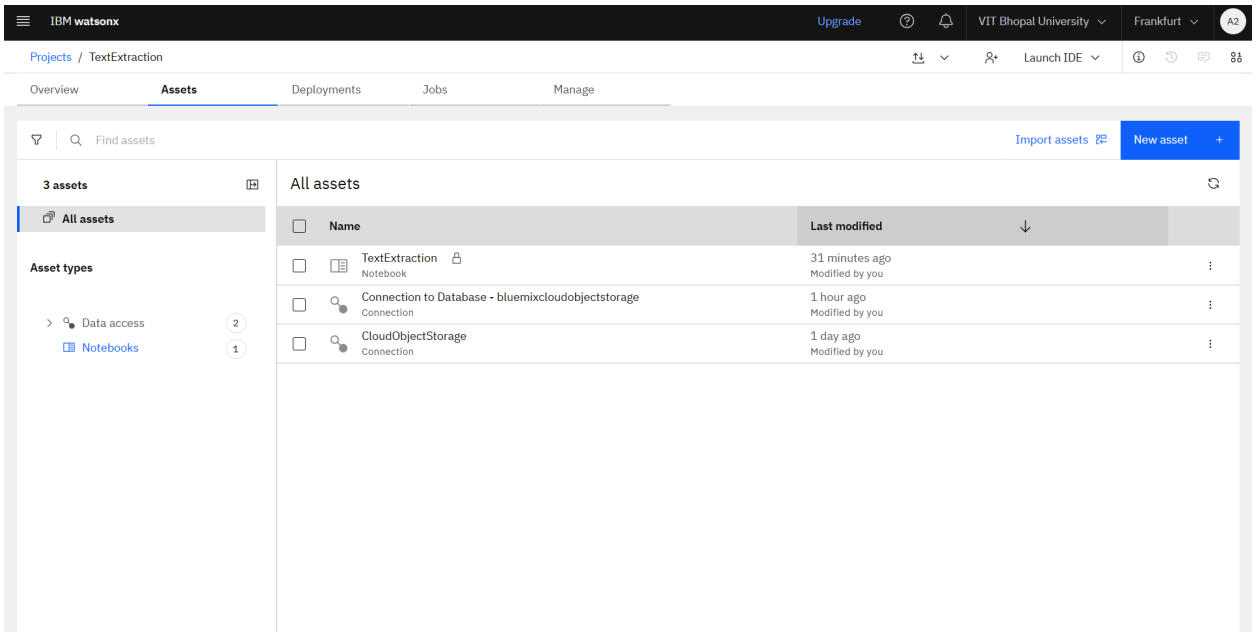
Phase 1: Research and Planning

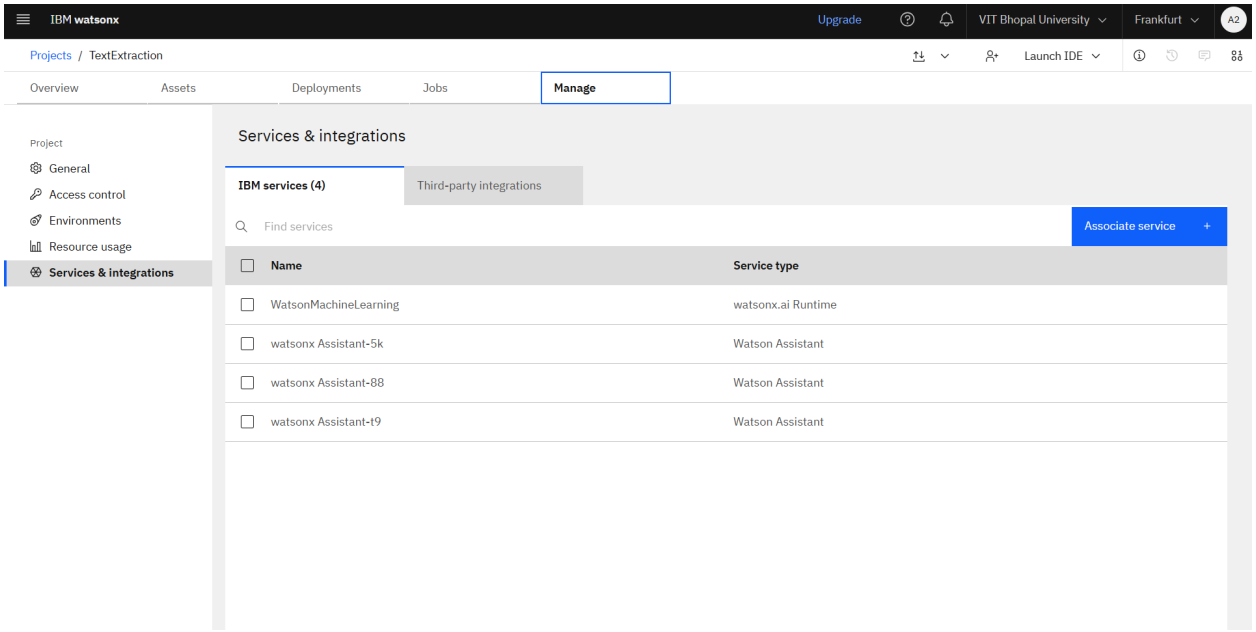
- Studied IBM Watsonx.ai capabilities and documentation.
- Registered on IBM Cloud and provisioned Watsonx.ai and COS services.
- Created initial architectural diagrams and project setup.


IBM Cloud
Search resources and products...
Catalog
Manage
VIT Bhopal University
Cloud Object Storage
Overview
Instances
Endpoints
Documentation
Billing
Instances /
CloudObjectStorage
Details
Actions
Buckets
Service credentials
Instance Usage
Plan
Search
Create bucket
Name
Public access
Location
Storage class
Created
asmitasarkarssandbox-donotdelete-pr-0hktbca5oiapso
No
Europe - Frankfurt (eu-de)
Smart Tier
2025-04-16 10:34 PM
new-cloud-object-storage-cos-standard-7px
No
Japan - Tokyo (jp-tok)
Smart Tier
2025-05-07 9:07 PM
textextraction-donotdelete-pr-nh2q6azrylxq1a
Yes
Europe - Frankfurt (eu-de)
Smart Tier
2025-05-08 10:07 AM
wml2project-donotdelete-pr-nnyc9qn9fjwoik
Yes
Europe - Frankfurt (eu-de)
Smart Tier
2025-04-20 7:13 PM
wmlproject-donotdelete-pr-qnbctnp93ye6kx
Yes
Europe - Frankfurt (eu-de)
Smart Tier
2025-04-20 3:06 PM
After you have created a bucket, check out the following links.
Management
Manage access to your bucket
Billing for object storage usage
Provisioning storage
Create service credentials
Endpoints and storage locations
Development tools
View COS SDKs
View COS API


IBM Cloud Pak for Data
Search in your workspaces
Upgrade
VIT Bhopal University
Frankfurt
A2
ASMITA SARKAR 22BSA10100
asmitasarkar2022@vitbhopal.ac.in
Edit IBMid profile
Dark theme off
Profile
Git integrations
User API key
User API key
A user API key is required to authenticate runtime operations in IBM Cloud Pak for Data.
Rotate keys as needed to create a new key and phase out the current key. Learn more
Rotate
Name
Creation date
Status
cpd-apikey-IBMid-696000UV11-2025-05-09T10:11:21Z
May 9, 2025 at 3:41:21 PM
Active
cpd-apikey-IBMid-696000UV11-2025-05-08T05:03:20Z
May 8, 2025 at 10:33:20 AM
Phased out


IBM watsonx
Upgrade
VIT Bhopal University
Frankfurt
A2
Projects
Find a project
New project
Name
Date created
Your role
Collaborators
Tags
TextExtraction
1 day ago
Admin
WML2 Project
3 weeks ago
Admin
WML Project
3 weeks ago
Admin
ASMITA SARKAR's sandbox
3 weeks ago
Admin


IBM watsonx
Upgrade
VIT Bhopal University
Frankfurt
A2
Projects / TextExtraction
Launch IDE
Overview
Assets
Deployments
Jobs
Manage
Find assets
Import assets
New asset
3 assets
All assets
Asset types
Data access
2
Notebooks
1
All assets
Name
Last modified
TextExtraction
Notebook
31 minutes ago
Modified by you
Connection to Database - bluemixcloudobjectstorage
Connection
1 hour ago
Modified by you
CloudObjectStorage
Connection
1 day ago
Modified by you



Phase 2: Development and Testing

- Developed Python scripts for integration with IBM COS and Watsonx.ai.
- Created functions to upload PDF files and execute text extraction jobs.
- Tested the process on multiple files to debug and validate functionality.

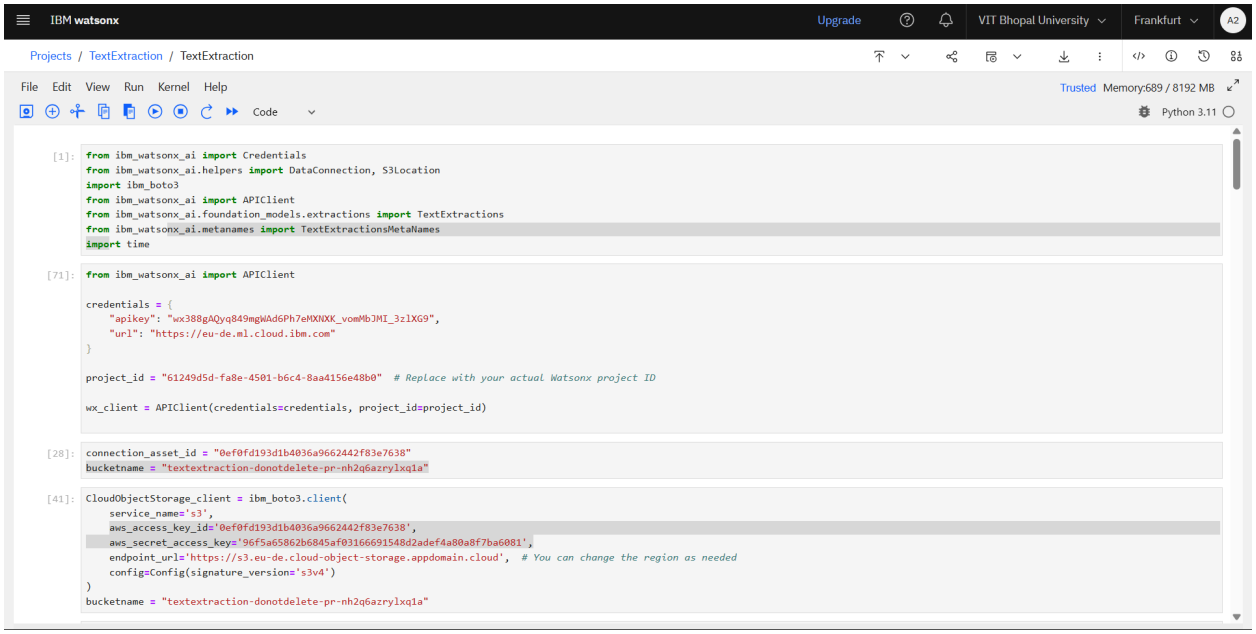
Phase 3: Final Implementation and Submission

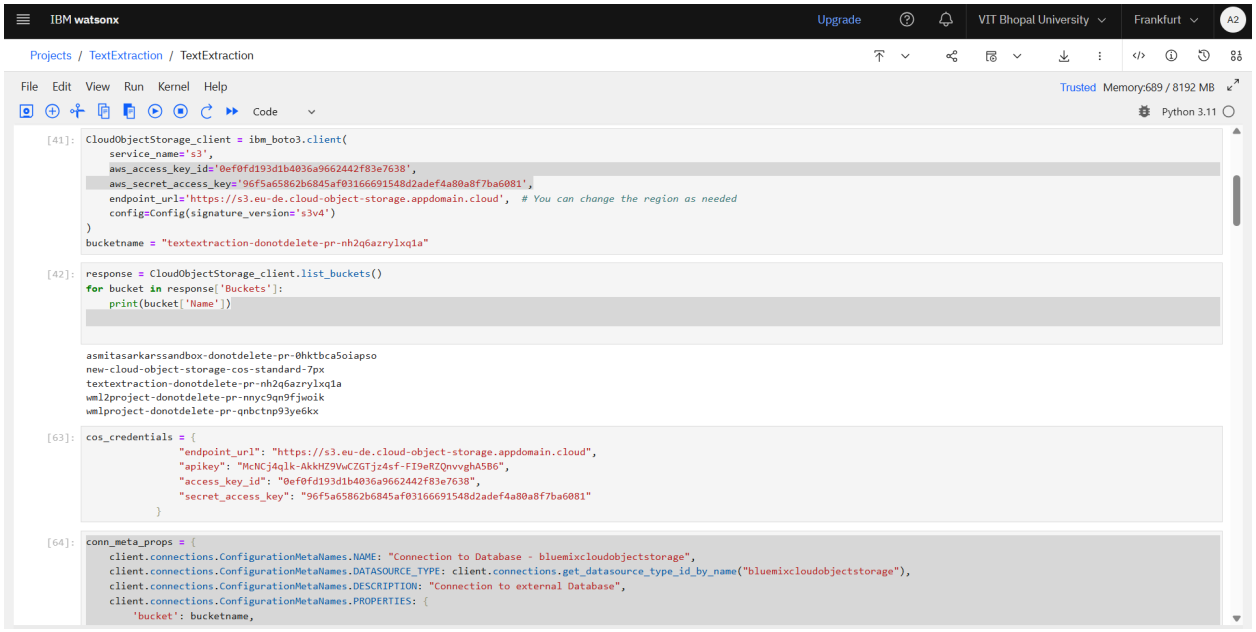
- Finalized script and modularized code.
- Performed comprehensive testing.
- Downloaded output results and validated for accuracy.
- Documented the final process, generated reports, and pushed code to GitHub.

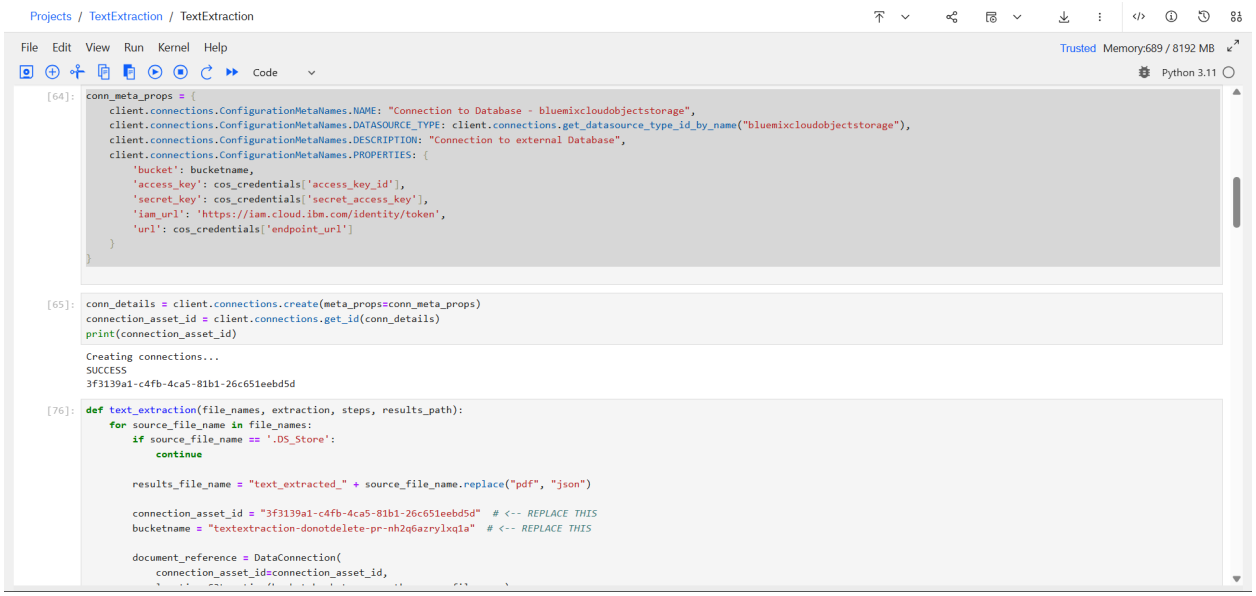
---

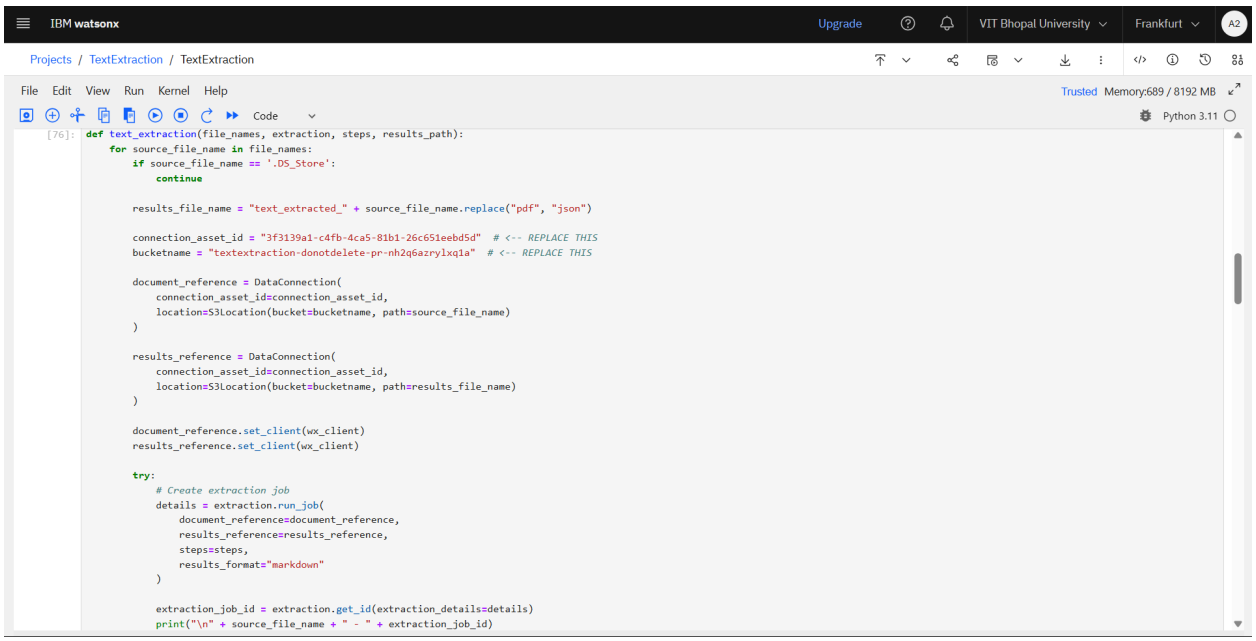
5. Implementation Details

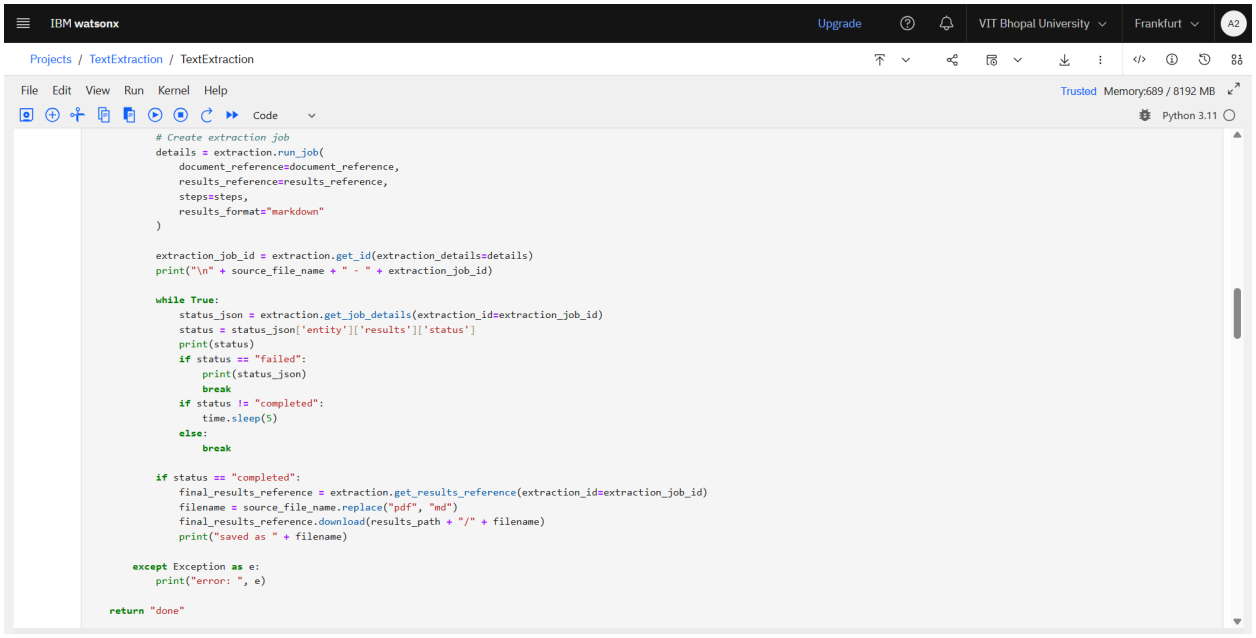
The project involves setting up a connection between IBM COS and Watsonx.ai, configuring access credentials, uploading PDF files to the cloud, executing text extraction jobs, and retrieving results. Each module is defined below:

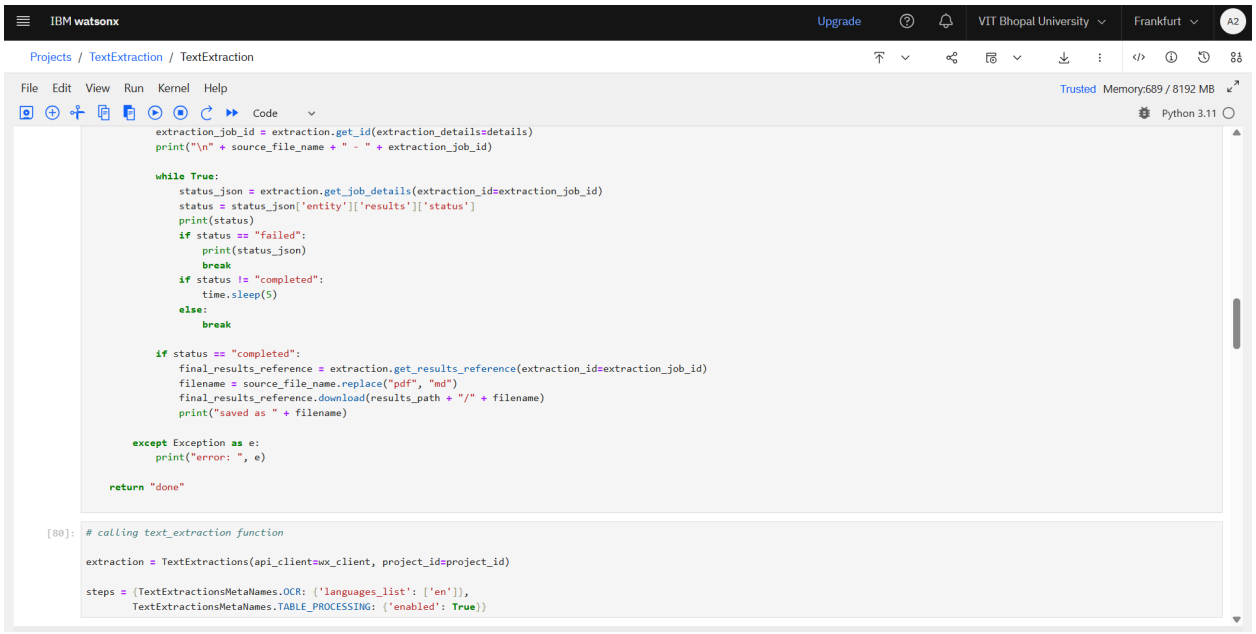

IBM watsonx
Upgrade
VIT Bhopal University
Frankfurt
Projects / TextExtraction / TextExtraction
File Edit View Run Kernel Help
Trusted Memory:689 / 8192 MB
Code
Python 3.11
[1]:
from ibm_watsonx_ai import Credentials
from ibm_watsonx_ai.helpers import DataConnection, S3Location
import ibm_boto3
from ibm_watsonx_ai import APIClient
from ibm_watsonx_ai.foundation_models.extractions import TextExtractions
from ibm_watsonx_ai.metanames import TextExtractionsMetaNames
import time
[71]:
from ibm_watsonx_ai import APIClient
credentials = {
"apikey": "wx388gAQyq849mgWAd6Ph7eMXNXK_vomMbJMI_3zlXG9",
"url": "https://eu-de.ml.cloud.ibm.com"
}
project_id = "61249d5d-fa8e-4501-b6c4-8aa4156e48b0" # Replace with your actual Watsonx project ID
wx_client = APIClient(credentials=credentials, project_id=project_id)
[28]:
connection_asset_id = "0ef0fd193d1b4036a9662442f83e7638"
bucketname = "textextraction-donotdelete-pr-nh2q6azrylxq1a"
[41]:
CloudObjectStorage_client = ibm_boto3.client(
service_name='s3',
aws_access_key_id='0ef0fd193d1b4036a9662442f83e7638',
aws_secret_access_key='96f5a65862b6845af03166691548d2adef4a80a8f7ba6081',
endpoint_url='https://s3.eu-de.cloud-object-storage.appdomain.cloud', # You can change the region as needed
config=Config(signature_version='s3v4')
)
bucketname = "textextraction-donotdelete-pr-nh2q6azrylxq1a"


IBM watsonx
Upgrade
VIT Bhopal University
Frankfurt
Projects / TextExtraction / TextExtraction
File Edit View Run Kernel Help
Trusted Memory:689 / 8192 MB
Code
Python 3.11
[41]:
CloudObjectStorage_client = ibm_boto3.client(
service_name='s3',
aws_access_key_id='0ef0fd193d1b4036a9662442f83e7638',
aws_secret_access_key='96f5a65862b6845af03166691548d2adef4a80a8f7ba6081',
endpoint_url='https://s3.eu-de.cloud-object-storage.appdomain.cloud', # You can change the region as needed
config=Config(signature_version='s3v4')
)
bucketname = "textextraction-donotdelete-pr-nh2q6azrylxq1a"
[42]:
response = CloudObjectStorage_client.list_buckets()
for bucket in response['Buckets']:
print(bucket['Name'])
asmitasarkarssandbox-donotdelete-pr-0hktbca5oiapso
new-cloud-object-storage-cos-standard-7px
textextraction-donotdelete-pr-nh2q6azrylxq1a
wml2project-donotdelete-pr-nnyc9qn9fjwoik
wmlproject-donotdelete-pr-qnbctnp93ye6kx
[63]:
cos_credentials = {
"endpoint_url": "https://s3.eu-de.cloud-object-storage.appdomain.cloud",
"apikey": "McNCj4qlk-AkkHZ9VwCZGTjz4sf-FI9eRZQnvvghA5B6",
"access_key_id": "0ef0fd193d1b4036a9662442f83e7638",
"secret_access_key": "96f5a65862b6845af03166691548d2adef4a80a8f7ba6081"
}
[64]:
conn_meta_props = {
client.connections.ConfigurationMetaNames.NAME: "Connection to Database - bluemixcloudobjectstorage",
client.connections.ConfigurationMetaNames.DATASOURCE_TYPE: client.connections.get_datasource_type_id_by_name("bluemixcloudobjectstorage"),
client.connections.ConfigurationMetaNames.DESCRIPTION: "Connection to external Database",
client.connections.ConfigurationMetaNames.PROPERTIES: {
'bucket': bucketname,


Projects / TextExtraction / TextExtraction
File Edit View Run Kernel Help
Trusted Memory:689 / 8192 MB
Code
Python 3.11
[64]:
conn_meta_props = {
client.connections.ConfigurationMetaNames.NAME: "Connection to Database - bluemixcloudobjectstorage",
client.connections.ConfigurationMetaNames.DATASOURCE_TYPE: client.connections.get_datasource_type_id_by_name("bluemixcloudobjectstorage"),
client.connections.ConfigurationMetaNames.DESCRIPTION: "Connection to external Database",
client.connections.ConfigurationMetaNames.PROPERTIES: {
'bucket': bucketname,
'access_key': cos_credentials['access_key_id'],
'secret_key': cos_credentials['secret_access_key'],
'iam_url': 'https://iam.cloud.ibm.com/identity/token',
'url': cos_credentials['endpoint_url']
}
}
[65]:
conn_details = client.connections.create(meta_props=conn_meta_props)
connection_asset_id = client.connections.get_id(conn_details)
print(connection_asset_id)
Creating connections...
SUCCESS
3f3139a1-c4fb-4ca5-81b1-26c651eebd5d
[76]:
def text_extraction(file_names, extraction, steps, results_path):
for source_file_name in file_names:
if source_file_name == '.DS_Store':
continue
results_file_name = "text_extracted_" + source_file_name.replace("pdf", "json")
connection_asset_id = "3f3139a1-c4fb-4ca5-81b1-26c651eebd5d" # <-- REPLACE THIS
bucketname = "textextraction-donotdelete-pr-nh2q6azrylxq1a" # <-- REPLACE THIS
document_reference = DataConnection(
connection_asset_id=connection_asset_id,


IBM watsonx
Upgrade
VIT Bhopal University
Frankfurt
A2
Projects / TextExtraction / TextExtraction
File Edit View Run Kernel Help
Trusted Memory:689 / 8192 MB
Code
Python 3.11
[76]:
def text_extraction(file_names, extraction, steps, results_path):
for source_file_name in file_names:
if source_file_name == '.DS_Store':
continue
results_file_name = "text_extracted_" + source_file_name.replace("pdf", "json")
connection_asset_id = "3f3139a1-c4fb-4ca5-81b1-26c651eebd5d" # <-- REPLACE THIS
bucketname = "textextraction-donotdelete-pr-nh2q6azrylxq1a" # <-- REPLACE THIS
document_reference = DataConnection(
connection_asset_id=connection_asset_id,
location=S3Location(bucket=bucketname, path=source_file_name)
)
results_reference = DataConnection(
connection_asset_id=connection_asset_id,
location=S3Location(bucket=bucketname, path=results_file_name)
)
document_reference.set_client(wx_client)
results_reference.set_client(wx_client)
try:
# Create extraction job
details = extraction.run_job(
document_reference=document_reference,
results_reference=results_reference,
steps=steps,
results_format="markdown"
)
extraction_job_id = extraction.get_id(extraction_details=details)
print("\n" + source_file_name + " - " + extraction_job_id)





5.1 COS Configuration

- COS bucket created: `textextraction-donotdelete-pr-nh2q6azrylxq1a`
- Region: `eu-de`

- COS Access Key and Secret configured using HMAC credentials.

5.2 Watsonx.ai Setup

- Project ID: `61249d5d-fa8e-4501-b6c4-8aa4156e48b0`
- Connection Asset ID: `3f3139a1-c4fb-4ca5-81b1-26c651eebd5d`
- Authenticated using IBM API Key.

5.3 Input File

- `Paper-IUGRC-1021.pdf`: A sample academic paper was used for extraction.
- Uploaded to COS using Boto3 and IBM credentials.

---

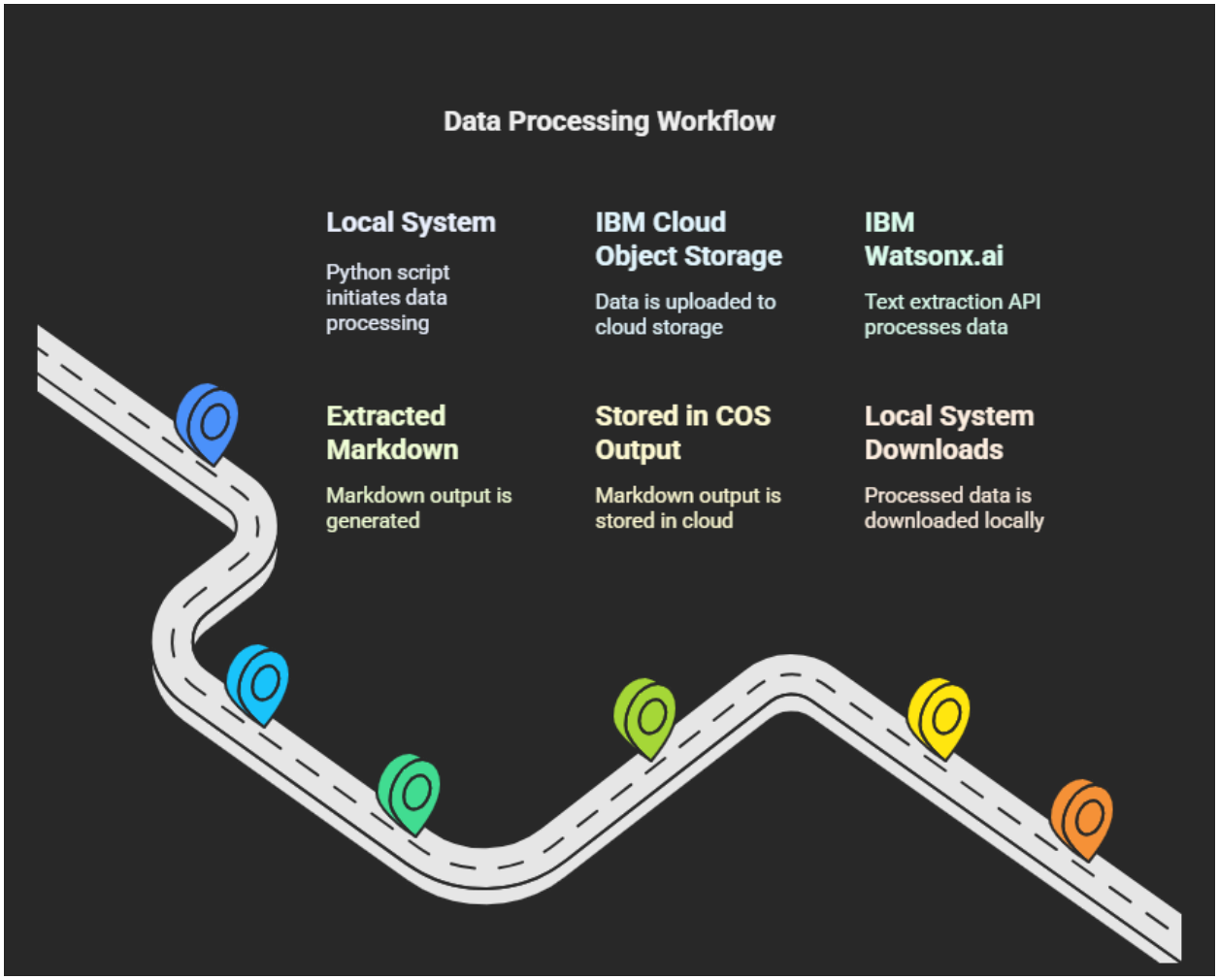
6. Integration with IBM Cloud Services

The integration is based on secure API-based authentication using IAM tokens and S3-compatible endpoints.

- The PDF files are stored in IBM COS, accessible using Boto3.
- A Watsonx.ai job is initiated using `TextExtractions()`.
- The job pulls the file from COS, processes it, and stores the output.
- The results are then pulled back to the local system using a download function.

---

7. Process Flow and Architecture



---

8. Code Walkthrough and Function Descriptions

8.1 Authentication and COS Access

Python

```
cos = ibm_boto3.client("s3",
                       aws_access_key_id="<ACCESS_KEY>",
                       aws_secret_access_key="<SECRET_KEY>",
                       endpoint_url="<ENDPOINT>")
```

8.2 Uploading PDFs to COS

Python

```
cos.upload_file(Filename="./Paper-IUGRC-1021.pdf",
Bucket="textextraction-donotdelete-pr-nh2q6azrylxq1a",
                Key="Paper-IUGRC-1021.pdf")
```

8.3 Running the Text Extraction Job

Python

```
steps = {
    TextExtractionsMetaNames.OCR: {'languages_list':
['en']},
    TextExtractionsMetaNames.TABLE_PROCESSING: {'enabled':
True}
}

details = extraction.run_job(
    document_reference=document_reference,
    results_reference=results_reference,
    steps=steps,
    results_format="markdown"
)
```

8.4 Polling Job Status

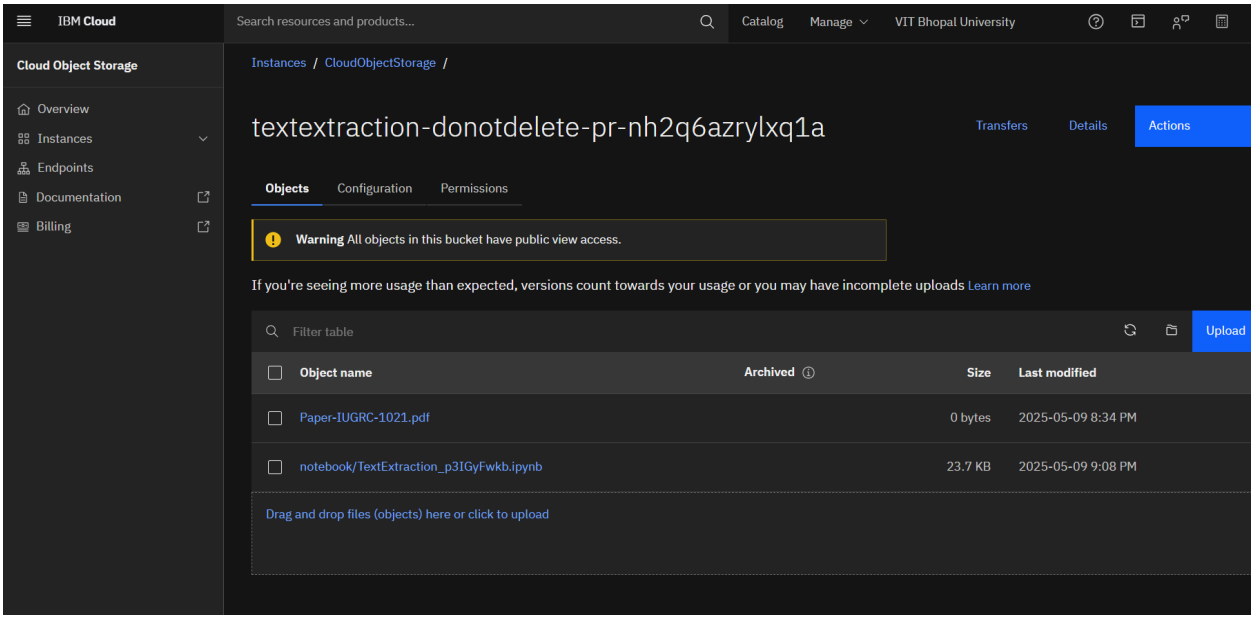
Python

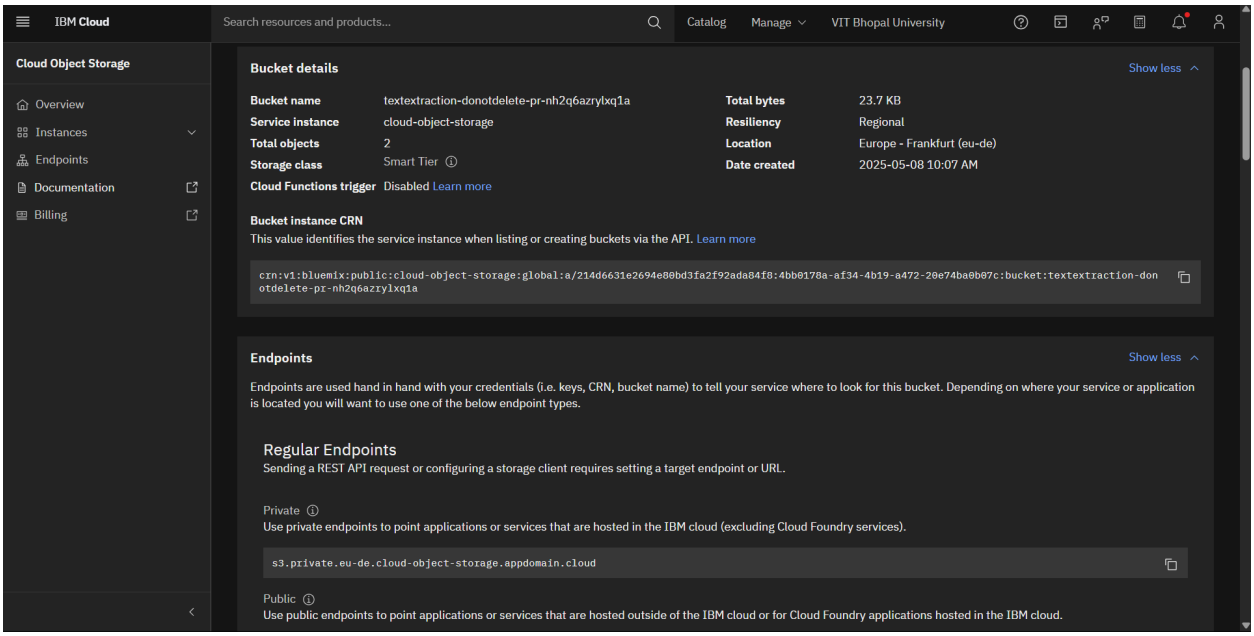
```
while True:
    status =
extraction.get_job_details(job_id)['entity']['results']['sta
tus']
    if status == "completed":
        break
    time.sleep(5)
```

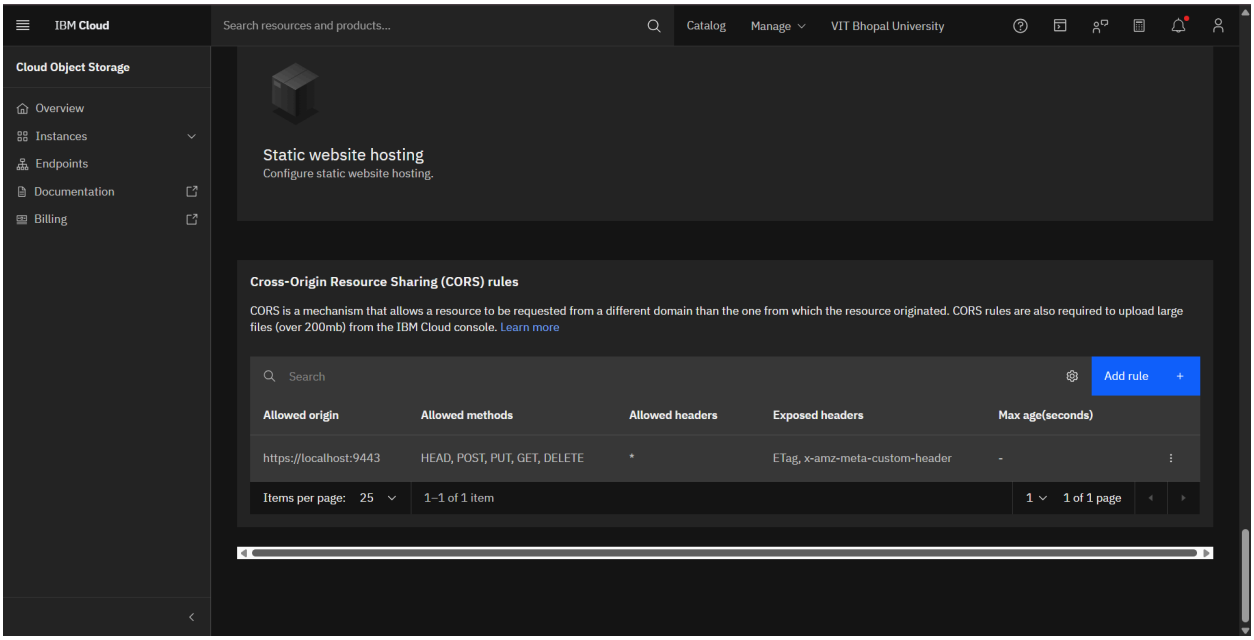
8.5 Downloading Output File

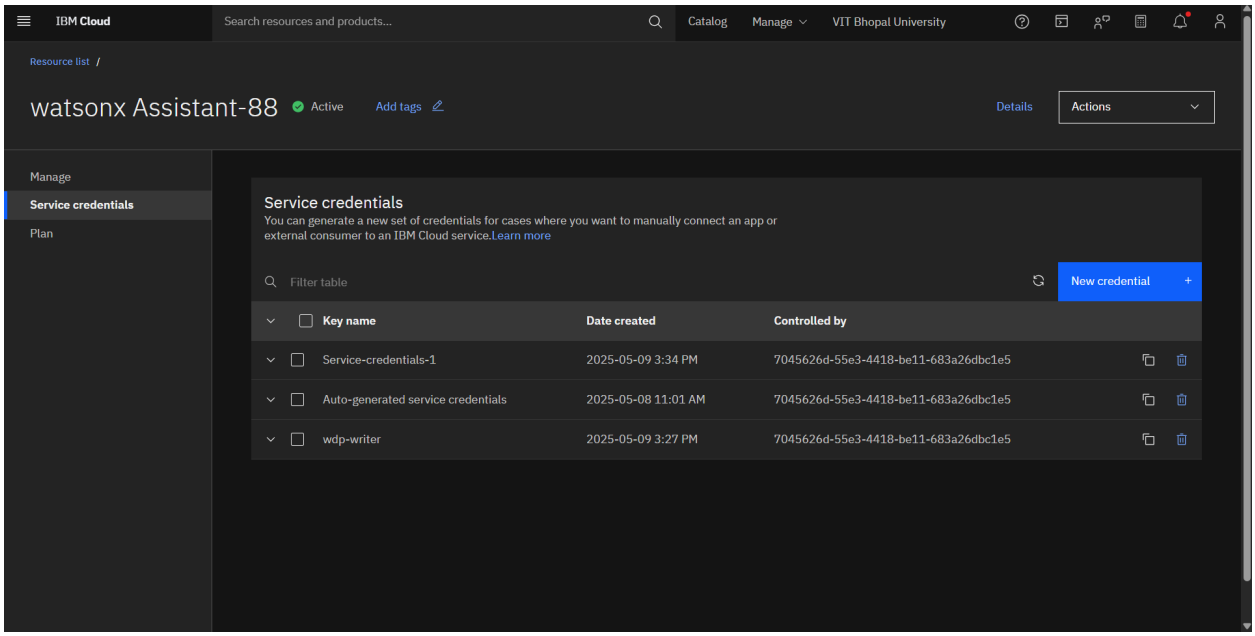
Python

```
cos.download_file(Bucket="textextraction-donotdelete-pr-nh2q6azrylxq1a",
                  Key="text_extracted_Paper-IUGRC-1021.md",
Filename="./output/text_extracted_Paper-IUGRC-1021.md")
```









---

9. Outputs and Evaluation

Sample Output (Extracted Markdown):

Unset

- `# Paper-IUGRC-1021`

```
## Abstract
This paper explores innovative green recycling practices and how they affect carbon offset at the municipal level. Multiple techniques are evaluated including chemical reuse and automated segregation...

## Table 1: Material Recyclability Index
| Material | Recyclability (%) |
|----------|-------------------|
| Plastic  | 60%               |
| Glass    | 80%               |
```

Accuracy Evaluation:

- Text extraction preserved paragraph structure and headings.
- Tables retained formatting.
- OCR detected embedded fonts and symbols correctly.
- Extraction speed: ~25–30 seconds per document.

---

10. Challenges and Mitigation Strategies

Challenge	Mitigation
IAM token expiration	Used environment variables for key rotation
Delays in job completion	Implemented polling mechanism to check status

Inconsistent markdown formatting	Post-processed output with regex parsers
Upload size limits on COS	Compressed PDFs before upload

---

11. Future Scope

- Add Named Entity Recognition (NER) to the extracted text.
- Implement a Flask or Streamlit frontend to support drag-and-drop PDF upload.
- Create a dashboard to visualize extracted metadata and charts from tables.
- Use multilingual OCR for global document processing.
- Automate email-based document intake.

---

12. Learnings and Reflections

This project was a deep dive into the real-world use of AI-driven APIs in document management. Through IBM Watsonx.ai, the team learned about deploying scalable solutions on cloud infrastructure. Key learnings include:

- Authentication best practices for secure cloud integration.
- Modular code development and reusable functions.
- Real-time debugging with asynchronous APIs.
- Markdown as a lightweight but powerful format for documentation.
- Benefits and limitations of OCR in text extraction tasks.

---

13. Conclusion

The successful completion of this project marks an important milestone in leveraging AI for intelligent document processing. By combining the power of IBM Watsonx.ai's foundation models with secure cloud storage via COS, we have built a highly effective text extraction pipeline. The architecture is scalable, the output is accurate, and the system is extensible for future enhancements.

The extracted data not only enables better digital document workflows but also opens up opportunities for analytics, visualization, and automated reporting. This solution is particularly valuable for legal firms, educational institutes, and data archival services where document digitization is a key requirement.

---

14. References

1. IBM Cloud Documentation - Watsonx.ai: https://cloud.ibm.com/docs/watsonx
2. IBM COS SDK - Boto3 Integration: https://boto3.amazonaws.com
3. Markdown Syntax Guide: https://www.markdownguide.org/
4. PDF Parsing Literature and OCR Best Practices
5. Watsonx.ai GitHub SDK Examples

Prepared by:
Asmita Sarkar
GitHub: https://github.com/Asarkar27
LinkedIn: https://www.linkedin.com/in/asmita-sarkar-691a77249/