

**Prepared by:** **Sinchana Ramesh | 14 June 2024**

DOCUMENT UNDERSTANDING(DU)

Contents

[1. INTRODUCTION 4](#_Toc171888056)

[1.1. What is Document Understanding ? 4](#_Toc171888057)

[1.2. Benefits of Document Understanding 4](#_Toc171888058)

[2. CONFIGURATIONS for DOCUMENT UNDERSTANDING 5](#_Toc171888059)

[3. PROCESS FLOW 8](#_Toc171888060)

[4. DU Service integrated within oracle INTEGRATION 9](#_Toc171888061)

[4.1. Connections 9](#_Toc171888062)

[4.2. Integration 11](#_Toc171888063)

[5. USE-CASE SCENARIO 19](#_Toc171888064)

[5.1. Use Case: Document Extraction and Classification 19](#_Toc171888065)

[6. Summary 21](#_Toc171888066)

Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Changed By | Reason for Change |
| 14/06/2024 | **1.0** |  |  |
|  |  |  |  |

Circulation List

|  |  |
| --- | --- |
| Name | Organisation/Title |
| Sinchana Ramesh | **Version1** |
|  |  |

Reference Documents

|  |  |  |
| --- | --- | --- |
| Title | Description | Owner |
|  |  |  |
|  |  |  |

# **INTRODUCTION**

## What is Document Understanding ?

In today's data-driven environment, organizations face the challenge of efficiently extracting important information from various documents. OCI Document Understanding(DU) is a cloud-native, serverless service that leverages deep learning-based models, both prebuilt and custom, accessible via REST APIs. It enables you to extract text and tables, recognise document types, and identify key fields from business documents such as receipts, invoices, and other documents.

The service can be accessed through the Oracle Cloud Console, OCI SDKs in Python and Java, or the OCI CLI.

Document Understanding integrates deep learning models, Optical Character Recognition (OCR), and Natural Language Processing (NLP), surpassing traditional document handling methods. It extracts and categorises text and tables, recognises document types, and identifies key fields from business documents. This enhances operational efficiency and accuracy. As organisations manage increasing volumes of information, OCI Document Understanding becomes an invaluable solution, streamlining processes, reducing errors, and optimising resource utilisation.

## Benefits of Document Understanding

* **Time Savings**: Automates the extraction of information from documents, saving time compared to manual data entry.
* **Reduced Errors**: Minimises the chances of human errors in data extraction by automating the process.
* **Increased Efficiency**: Streamlines document processing, leading to more efficient workflows and quicker decision-making.
* **Enhanced Accuracy**: Leverages deep learning models to provide precise and reliable data extraction and document classification.
* **No Expertise Required**: Utilises prebuilt or custom features without the need for any data science experience.
* **Versatile Access**: Extracted data is readily available, enabling fast and easy retrieval for analysis or decision-making.
* **Adaptability to Document Variations**: DU systems can adapt to different document formats and layouts, accommodating variations without compromising accuracy.

# **CONFIGURATIONS for DOCUMENT UNDERSTANDING**

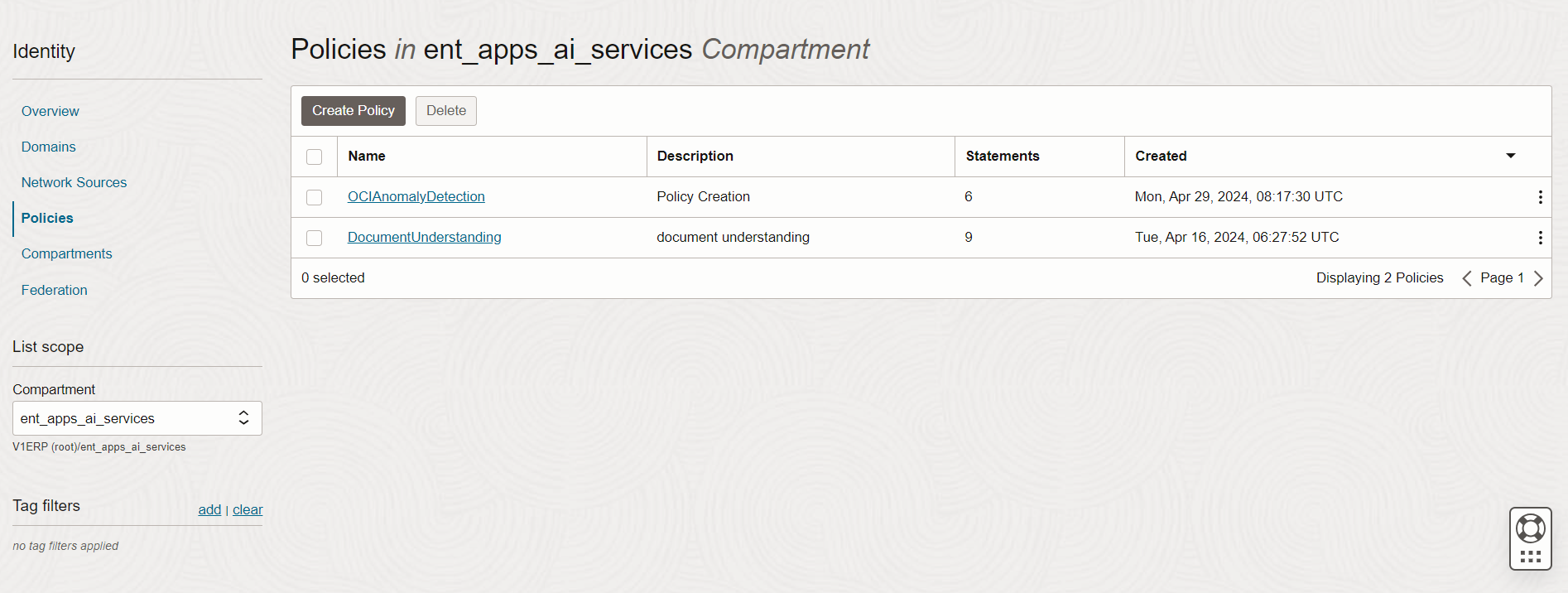
**Step 1 : Setting up the Policies.**

* To create a new policy, navigate to : Identity & Security > Identity > Policies as shown in the below image.
* *Note : Any user can create a new policy; no specific role or permissions are required*.



*Figure 1: Navigation to Policies tab*

* Click on **Policies** to be redirected to the page below in the image,
* Select your desired Compartment which is a prerequisite; you need to have a compartment defined.
* Click on **Create Policy** to create a new policy.



*Figure 2: Policy creation page*

* Provide the name and description of the policy and choose the compartment.
* Enable the show manual editor button and run the below commands and click on **Create**.
* Allow group 'OracleIdentityCloudService'/'DocumentUsers' to manage instance-family in compartment ent\_apps\_ai\_services
* Allow group 'OracleIdentityCloudService'/'DocumentUsers' to read app-catalog-listing in compartment ent\_apps\_ai\_services
* Allow group 'OracleIdentityCloudService'/'DocumentUsers' to use volume-family in compartment ent\_apps\_ai\_services
* Allow group 'OracleIdentityCloudService'/'DocumentUsers' to use virtual-network-family in compartment ent\_apps\_ai\_services
* Allow group 'OracleIdentityCloudService'/'DocumentUsers' to manage objects in compartment ent\_apps\_ai\_services
* Allow group 'OracleIdentityCloudService'/'DocumentUsers' to manage buckets in compartment ent\_apps\_ai\_services
* Allow group 'OracleIdentityCloudService'/'DocumentUsers' to manage data-labeling-family in compartment ent\_apps\_ai\_services
* Allow group 'OracleIdentityCloudService'/'DocumentUsers' to read buckets in compartment ent\_apps\_ai\_services
* Allow group 'OracleIdentityCloudService'/'DocumentUsers' to read objectstorage-namespaces in compartment ent\_apps\_ai\_services

A screenshot of a computer

Description automatically generated

Run the commands here

*Figure 3: Policy Tab*

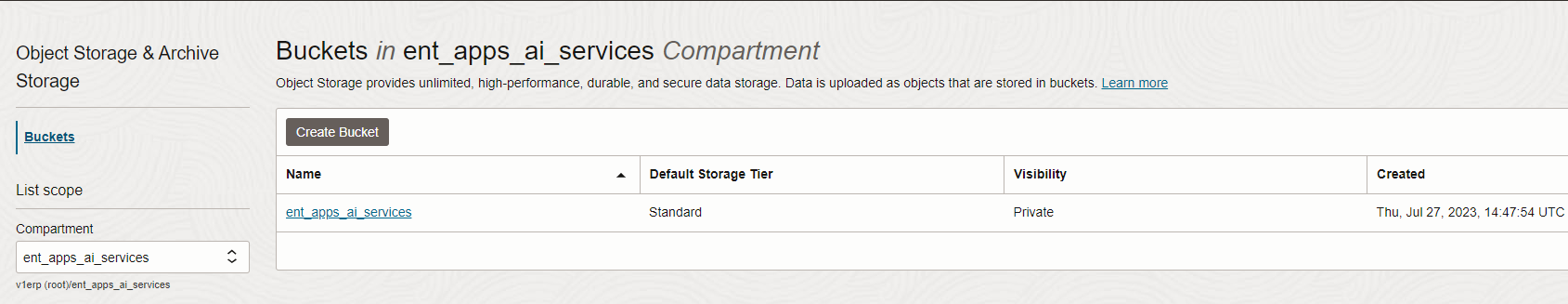
**Step 2 : Seamless Bucket Creation.**

* Navigate to Storage> Object Storage & Archive Storage > Buckets.
* Buckets are containers used to store and organise files within object storage systems. They help manage and categorize data, making it easy to access and retrieve files as needed.

A screenshot of a computer

Description automatically generated *Figure 4: Navigation tab for buckets*

* Click on **Buckets** to be redirected to the page below in the image,
* Select the desired Compartment which is a prerequisite; you need to have a compartment defined.
* Click **'Create Bucket**,' and provide the name and description.



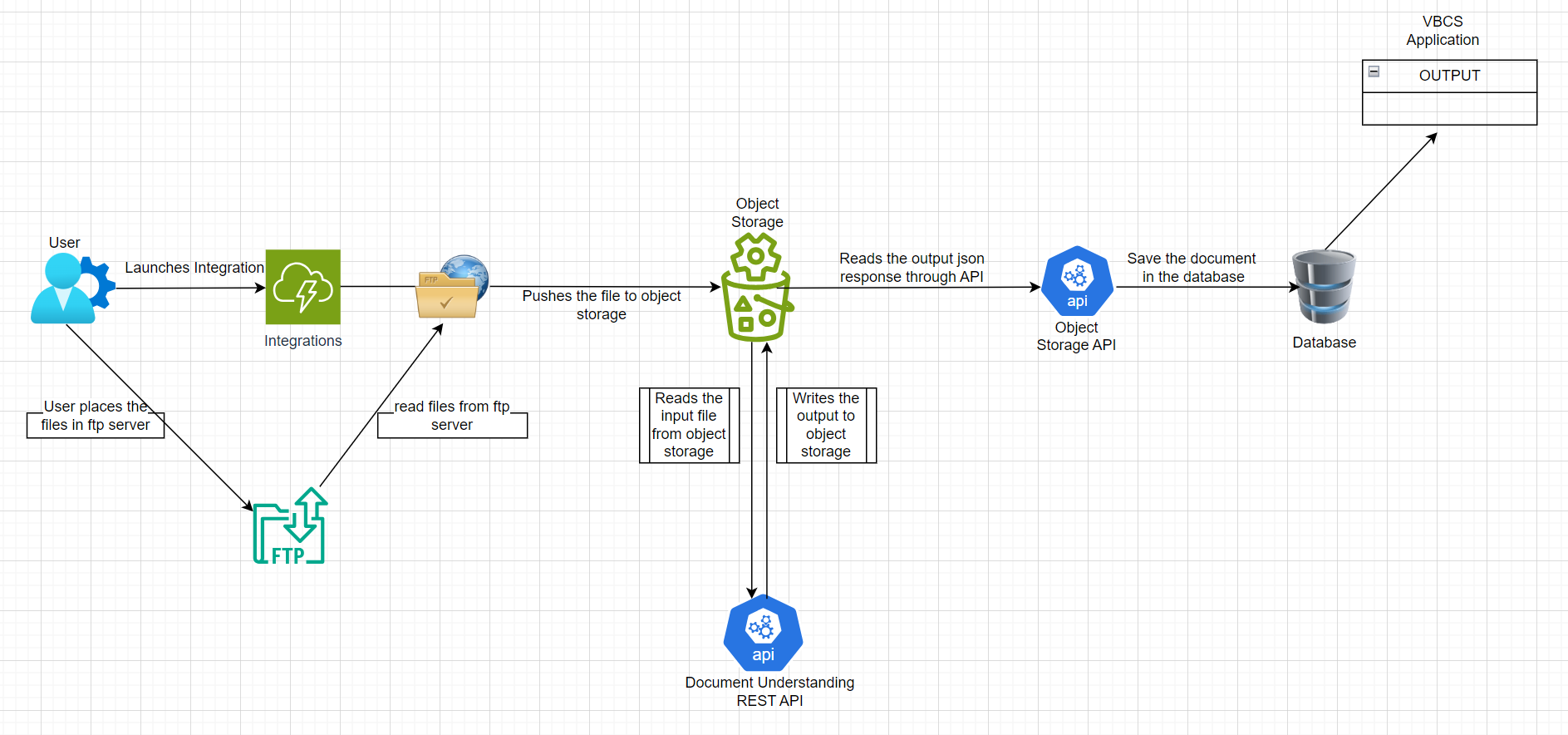
*Figure 5: Navigation to bucket creation*

# PROCESS FLOW

This section outlines the process flow diagram for integrating Document Understanding into

Oracle Integration. The sole purpose of using integration is to reduce manual intervention,

minimize errors, enhance efficiency and process multiple documents simultaneously.

 *Figure 6: Process Flow*

# DU Service integrated within oracle INTEGRATION

This section outlines how we can use the Oracle Document Understanding service offering from Oracle Cloud Infrastructure along with Oracle Integration. The purpose of using integration is to reduce manual intervention, minimize errors, and enhance efficiency. It enables sending multiple documents simultaneously and provides the option to schedule and run processes as needed.

We will be using the following services to implement,

1. Oracle Integration Standard Edition
2. Oracle Object Storage
3. Oracle Document Understanding Service REST APIs
4. SFTP
5. Oracle Database
6. Oracle Visual Builder Cloud Service

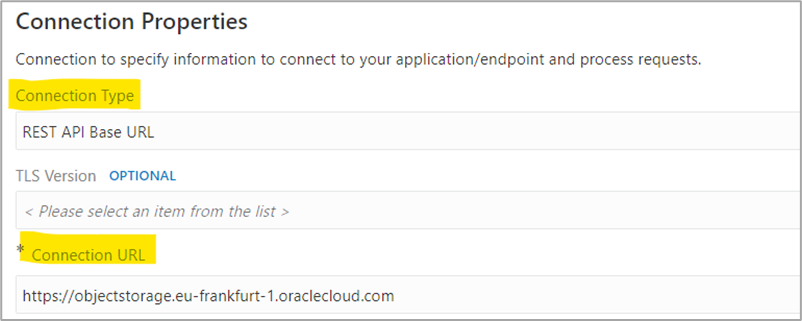
*Note: The Object Storage bucket and Document Understanding Service REST API must be in the same region.*

*For example, if you are using https://document.aiservice.ap-sydney-1.oci.oraclecloud.com then make sure the Object Storage bucket is also in the Sydney region.*

## Connections

**Step 1:** Configure the Oracle Integration Instance to invoke the Object Storage Rest API.

1. Create a connection to the Object Storage using the REST Connection type adapter.



A screenshot of a computer

Description automatically generated

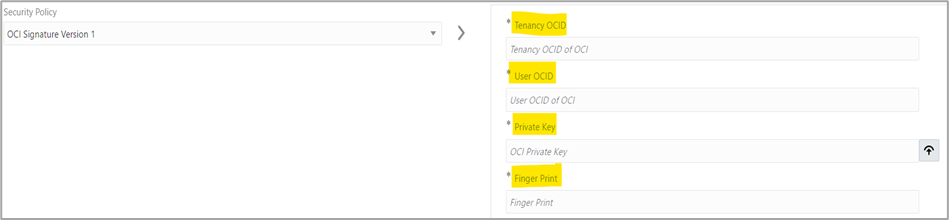
*Figure 7: Object Storage Rest API*

**Step 2:** Configure the Oracle Integration Instance to invoke the Document Understanding Rest API.

1. Create a connection to the Document Understanding Service using the REST Connection type adapter.

A screenshot of a computer

Description automatically generated

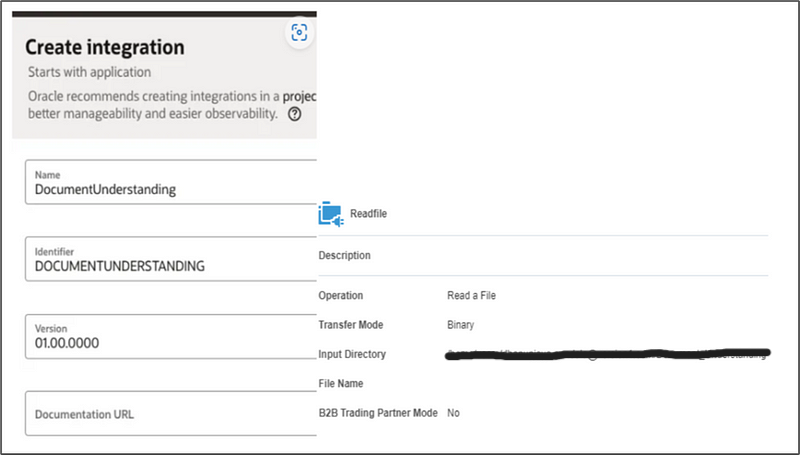


*Figure 8: Document Understanding REST API Connection*

## Integration

Create the integration to invoke the Document Understanding Service.

1. Insert a read operation to access all the files in the SFTP server as shown below.



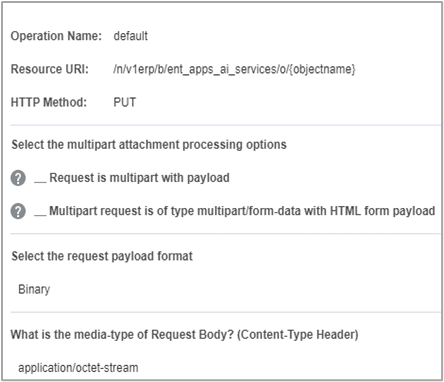
A white background with a couple of arrows

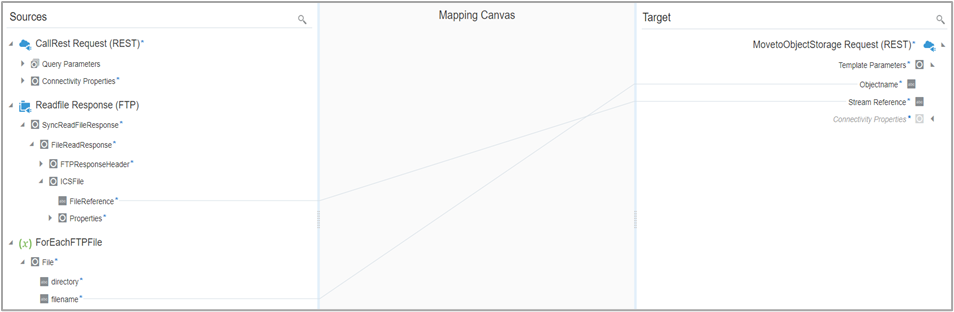
Description automatically generated with medium confidence

*Figure 9: Invoking Read operation and mapping*

2. Insert an Invoke operation to the Object Storage Connection as shown below.

* This operation reads the files placed on the SFTP server and pushes them to object storage. Storing the files here allows the Document Understanding service to easily access them.

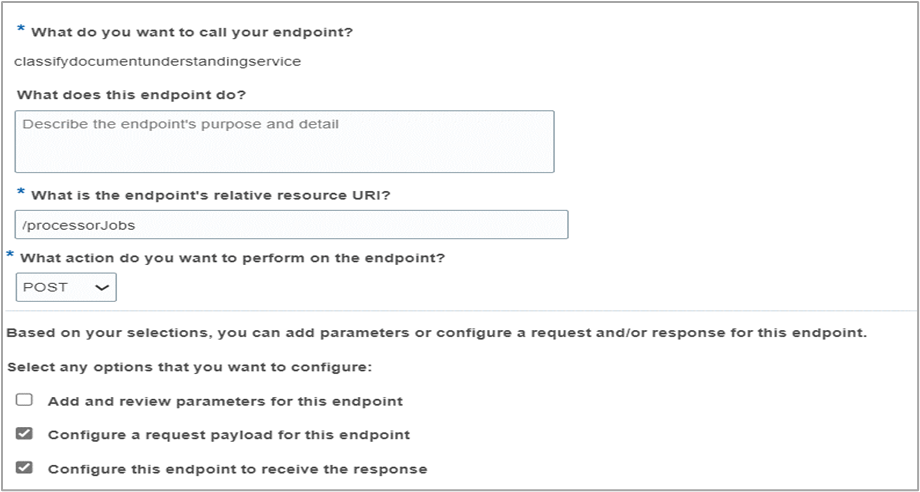




*Figure 10: Invoking Object Storage Rest API mapping*

3. Configure an Invoke operation to the Document Understanding Service.

* This operation triggers the document understanding REST API to process files from object storage.
* It classifies the document type and extracts information in key-value pair.
* Finally, the JSON response is sent back to object storage.



A screenshot of a computer

Description automatically generated

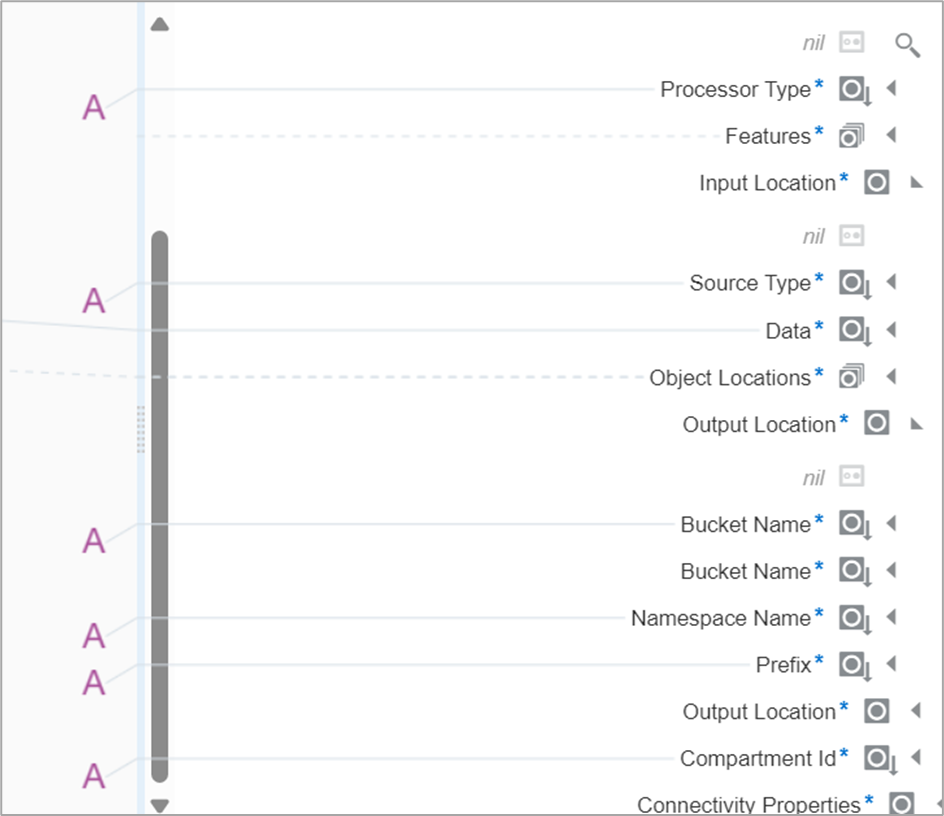
*I am using the below sample JSON input for the request payload.*

*Note : Please provide appropriate Compartment-Id, Namespace-name, Bucket-name, and Object-name.*

{  
 "processorConfig" : {  
 "processorType" : "GENERAL",  
 "features" : [ {  
 "featureType" : "DOCUMENT\_CLASSIFICATION",  
 "featureType" : "KEY\_VALUE\_EXTRACTION",  
 "maxResults" : 5  
 } ],  
 "language" : "ENG"  
 },  
 "inputLocation" : {  
 "sourceType" : "OBJECT\_STORAGE\_LOCATIONS",  
 "objectLocations" : [ {  
 "source" : "OBJECT\_STORAGE",  
 "namespaceName" : "abc",  
 "bucketName" : "demo",  
 "objectName" : "image.png"  
 } ]  
 },  
 "compartmentId" : "123456",  
 "outputLocation" : {  
 "namespaceName" : "abc",  
 "bucketName" : "demo",  
 "prefix" : "result"  
 }  
}

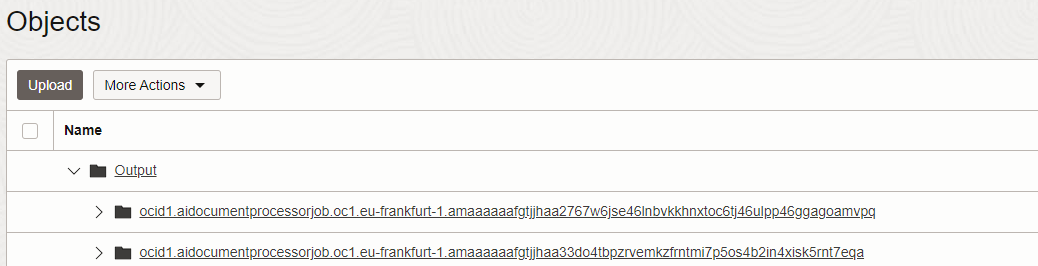
*Figure 11: Sample Json Code*

* Map the values for input variables, you can provide these values as part of your integration’s request input and map them dynamically.



*Figure 12: Mapping values for input variables*

* The output is saved to the same object storage bucket with the given prefix from the JSON payload.



*Figure 13: Sample output location*

4. Insert an invoke operation to read the JSON response from object storage using the Object Storage REST API, format the response into text fields, and insert the data into a temporary table, follow these steps.

* Use the Object Storage REST API to read the JSON response from object storage.
* Format the JSON response into text fields.
* Insert the formatted data into a temporary table for storing the information.

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A white background with lines

Description automatically generated

*Figure 14: Mapping for table insertion*

5. Insert an invoke operation to transfer data from the temporary table to the source table, call a PL/SQL procedure to insert data into the respective document tables based on the document type.

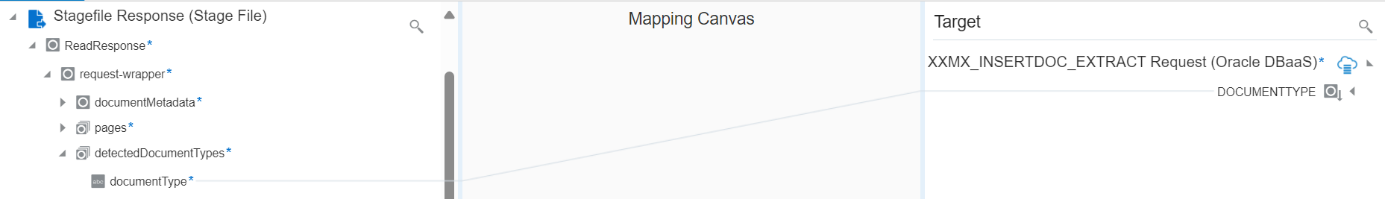
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* Map the value to the document type as shown below.



*Figure 15: Mapping for source table insertion*

6. Insert an archival operation to move the files into archival folder in SFTP server.

A screenshot of a computer

Description automatically generated

*Figure 16: Archival move operation*

7. Once all the steps are done, we are now ready to activate the integration and test.

* To expand this use case, we link the integration to a VBCS Application to initiate the process.

A screenshot of a computer

Description automatically generated

*Figure 17: VBCS Application*

# USE-CASE SCENARIO

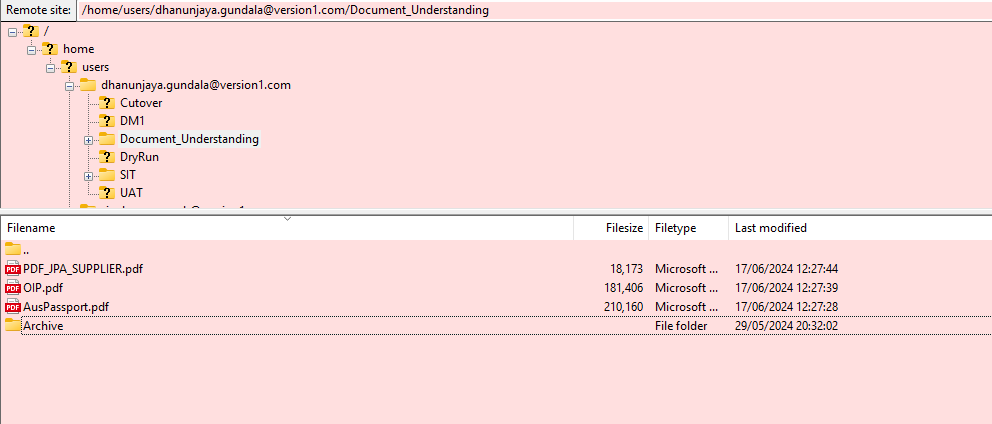
## Use Case: Document Extraction and Classification

**Scenario:**

When a user uploads documents, Document Understanding automatically categorizes each document type and extracts relevant details into structured key-value pairs. This streamlined process reduces manual effort and ensures accurate data extraction, enhancing overall efficiency in document management and information retrieval tasks.

#### Step 1: Document Integration

* To test the integration, place the documents in the SFTP Server used in the integration.

 *Figure 18: SFTP Server*

#### Step 2: Initiating the Process

* This is the VBCS page, featuring a button to extract documents and multiple tables to display the output in a tabular format based on the document type.

A screenshot of a computer

Description automatically generated

*Figure 19: VBCS Page*

* Click the button to extract, this action launches the integration process, which runs in the background. Once completed, the output will be displayed in the tables, showing the document type, and extracted information in key-value pairs.

A screenshot of a computer

Description automatically generated

*Figure 20: Invoice Tab*

A screenshot of a computer

Description automatically generated

*Figure 21: Receipt Tab*

A screenshot of a computer

Description automatically generated

*Figure 22: Passport Tab*

* The use case can be further tested with documents such as bank statements, driving license, resume, and payslip with formats like PDF, PNG, and JPEG.
* There is also an option to create custom models and train them for your specific document types.

# Summary

Document Understanding involves extracting and converting meaningful information from unstructured or semi-structured documents into structured data for analysis. This process utilizes machine learning, NLP and traditional OCR to automate information extraction. This overview provides a comprehensive look at document understanding, emphasizing its significance, the technologies it employs, and its practical applications. Whether you’re a business professional looking to streamline operations or a tech enthusiast curious about the latest advancements, understanding document understanding is crucial in harnessing the power of data.

A blue triangle with a white background

Description automatically generated

**Thank you**