Oracle Banking Trade Finance Process Management

Gen AI Service Installation Guide

Release 14.7.4.0.0

F99447-01

June 2024

ORACLE®

Oracle Banking Trade Finance Process Management Gen AI Service Installation Guide, Release 14.7.4.0.0

F99447-01

Contents

Preface

1   Introduction

2   Application Installation

3   Starting The Application

Index

Preface

Purpose

This guide provides information about the configurations of Generation Artifical Intellegence for OracleWebLogic application server.

Audience

This document is intended for WebLogic admin or ops-web team who are responsible for installing the OFSS banking products.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1
Introduction

This topic describes the systematic instructions to install Generative AI service.

Gen AI Service is an optional component that is shipped with various products that utilizes Generative AI for managing documents.

> **Note:**
>
> This service is available for on-premise products only.

Prerequisite

Follow the below steps to setup the prerequisites of Gen AI service.

1. The Operating System version should be same as product i.e., Oracle Linux 8.

   > **Note:**
   >
   > Other Oracle Linux OS are not supported due to incompatibility with gcc++ v17 compiler.

2. gcc++ compiler v17

   > **Note:**
   >
   > This compiler is required for tesseract.

3. Install the following OS packages:
   a. yum install zlib zlib-devel
   b. yum install libffi-devel openssl-devel
   c. yum install bzip2 bzip2-devel
   d. yum install poppler-utils
   e. yum install xz xz-devel xz-libs
   f. yum install mesa-libGL
   g. yum install mesa-libgbm
   h. yum install mesa-libglapi
   i. yum install sqlite-devel

4. Python Version: 3.9.5

5. Tesseract: 5.1.0

6. Document Verification Service: This is provided as part of the distribution.

7. OCI Credentials and configuration setup. Refer to OCI Credentials and Configuration Setup (page 2-5) to setup and configure OCI.

To run the Gen AI document analyzer service; Python, Tesseract, and Document Verification Service must be installed. Refer to **Document Verification Framework** section in ***Common Core Services Installation Guide*** to manually install these packages.

2
Application Installation

This topic provides the information to install the Gen AI application.

Gen AI document analyzer is a python-based application. The application is shipped as a byte-coded whl file. This wheel file installs all the implementation files without the dependencies. All the required dependencies are to be installed separately. It is recommended to install the whl file and the dependencies in a new virtual environment using **pip** so that it doesn't affect any other operations or applications running in the system.

Follow the steps below to install the app and the dependencies:

1. Use the below command to install the application wheel package provided, e.g.

   ```
   cmc_ml_genai_doc_analyzer-{version}-py3-none-any.whl
   ```

   ```
   pip install <wheel_package_name>.whl
   ```

2. All the dependencies must be installed.

2.1 Dependency Installation

This topic describes the information to install the dependencies.

After installing the Document verification service, the following dependencies must be installed. Please install the below third-party dependencies before starting the services.

> ✎ **Note:**
>
> These packages must be installed in the environment where the document verification services are installed.

Common Dependencies:

Install the below dependencies for all the LLM's.

> ✎ **Note:**
>
> These dependencies are mandatory.

> ✎ **Note:**
>
> If you are installing on server behind the proxy server then provide the proxy settings e.g., if connected to Oracle network then following proxy will work. Set the proxy before installing the dependencies according to the shell (bash/csh/sh etc).

https_proxy = http://www-proxy.us.example.com:80

https_proxy = http://www-proxy.idc.example.com:80

no_proxy=localhost,127.0.0.1,.us.example.com,.ocir.io,.examplecorp.com,.examplevcn.com

```
pip install pypdf==3.9.1
pip install PyPDF2==3.0.1
pip install Flask==2.2.5
pip install pdfminer.six==20221105
pip install pydantic==1.10.13
pip install pytesseract==0.3.9
pip install openpyxl==3.1.2
pip install PyMuPDF==1.22.5
pip install sentence-transformers==2.2.2
pip install tabulate==0.9.0
pip install oci==2.112.1
pip install oracledb==1.3.2
pip install langchain==0.1.13
pip install langchain-community==0.0.29
pip install langchain-core==0.1.42
pip install langchain-experimental==0.0.55
pip install langchain-text-splitters==0.0.1
pip install langsmith==0.1.37
pip install docx2txt==0.8
pip install tiktoken==0.5.2
pip install py-eureka-client==0.10.0
pip install pandas==1.4.1
pip install importlib-metadata==6.0.0
pip install xlrd==2.0.1
```

LLM Dependencies:

The dependencies mentioned below can be installed based on which large language model you choose to configure.

OpenAI LLM Dependencies:
Install the below dependencies, to use OpenAI LLM.

```
pip install openai==1.17.0
pip install langchain-openai==0.1.3
```

Cohere LLM Dependencies:
Install the below dependencies, to use Cohere LLM.

```
pip install cohere==5.1.7
pip install langchain-cohere==0.1.0
```

Gemini LLM Dependencies:
Install the below dependencies, to use Gemini LLM.

```
pip install google-generativeai==0.4.1
pip install langchain-google-genai==1.0.1
```

LLAMA LLM Dependencies:

Install the below dependencies, to use Llama LLM.

```
pip install llama-cpp-python==0.1.83
```

> **Note:**
>
> This application works when the above libraries are installed with required versions. Don't upgrade the libraries unless instructed in the documentation.

2.2 Configuration Update

This topic provides the information to update the configurations.

The following are the three configuration files provided:

1. **application-config.json**: This file contains the configuration details that are supposed to be changed by the user for each installation.

2. **system-config.json**: This file contains the internal settings of LLM. This should not be changed unless otherwise specified.

3. **logging-config.json**: This file is used for changing the logging settings. Use the default setting unless otherwise specified.

application-config.json:

Refer the below table for the fields and description of the fields:

Table 2-1   application-config.json

Parameter	Description
APPLICATION_NAME	Application name of the service to register on eureka.
LLM	Name of the LLM that you want to use. (openai, or cohere, or gemini). If application needs to be use multiple LLM's then install same app for each LLM and change this value to reflect each LLM. > **Note:** > One LLM can be used in one instance only.
LLM_API_KEY	The valid API Key of above mentioned LLM.
USE_CONFIG_LLM_API_KEY	Whether the API key should be used from this application config or not.
EXTRACTOR_TYPE	Whether you want to use OCI Document Understanding Service or the OFSS Document Verification Service for extraction
DELETE_AFTER_TRAINING	Whether the documents, trained files should be deleted post training or not.
WORKING_DOCUMENT_DIR	Path to local folder where trained files will be stored. User should have Read-Write permissions to this folder.

Table 2-1   (Cont.) application-config.json

Parameter	Description
USE_CLASSIFIER	Whether or not to use the LLM based classifier for classifying the documents.
CLASSIFIER_PARENT_DIR	Path to the parent directory for the documents required to use the classifier. Needed only if USE_CLASSIFIER is set to **yes**.
CLASSIFIER_MODEL_ID	Model ID of the custom trained classifier model. Needed only if USE_CLASSIFIER is set to **yes**.
OCI_CONFIG_FILE	Path to **oci_config.txt** file. You can get the file path after completion of OCI Credentials and configuration explained in OCI Credentials and Configuration Setup (page 2-5).
EUREKA_CLIENT_SERVICE_DEFAULT_ZONE	Address of Eureka for Service Discovery that will be used to connect with DMS (cmc-document-services) service.
DMS_DOWNLOAD_ENDPOINT	Endpoint of cmc-document-services used for downloading from DMS. For Example: /cmc-document-services/service/v1/documents?documentRefId=
DMS_UPLOAD_ENDPOINT	Endpoint of cmc-document-services used for uploading to DMS. For Example: /cmc-document-services/service/v1/documents DMS service is accessed by using this endpoint.
DMS_SERVICE	Name of the DMS (cmc-document-services) service to locate on eureka. For Example: cmc-document-services
SSL_CERTIFICATE	Path to your SSL Certificate (.crt file)
SSL_KEY	Path to your SSL Key (.key file)

2.3 Document Classifier

This topic provides the information on document classification.

To index the documents correctly, classification of them needs to be carried out. The application supports two ways of classifying the document, either it can look for reference in the file name or it can work with ML based classifier to classify the documents.

For Example: The document type can be specified with either using the file names as the keyword such as **statement** for bank statements, **paystub** for salary slips and **personal_doc** for personal documents or the user can opt to use the document classifier provided by us which can predict the document type using the LLM.

To use the document classifier, the user needs to specify a CLASSIFIER_PARENT_DIR in the **application-config.json** file. This path will be used as the location to store all the training data to train the classifier model. To train the model user needs to hit the endpoint: ***/docGpt/docClassify/train***.

Following is the body of the request:

```
{
"trainDir": "YOUR-FOLDER-NAME-WITH-TRAINING-DATA",
"llmAPIKey": "YOUR-LLM-API-KEY",
"llm": "YOUR-LLM-NAME"
}
```

The above request results in the training data to be uploaded in the **trainDir** folder at the *CLASSIFIER_PARENT_DIR* path. Inside the **trainDir** folder the folder structure should be in

the following format with the **country name** followed by the **document type**. Follow the recommended folder structure as given below:

CLASSIFIER_PARENT_DIR > trainDir > country name > document type .

Below is an example showing the contents of the trainDir folder.

Once the model is trained and the model ID is returned, this model ID has to be provided in the CLASSIFIER_MODEL_ID in the **application-config.json** file for inference. After this the model will be used internally by other APIs to classify a particular document and do QnA on top of it.

> **Note:**
>
> This feature is currently supported by cohere only.

If the document classifier is not used, the system uses the associated metadata of the uploaded files coming from the DMS to classify the documents for further training and querying. For these the **documentCatID** is used. A separate config file called **docCategory-config.json** is maintained to specify which documentCatIDs fall in which categories. Below is an example for the same:

```
{
"PERSONAL_DOC": ["passport", "dl", "pan", "voter", "aadhaar", "birth"],
"SALARY_SLIP": ["paystub", "salary"],
"BANK_STATEMENT": ["bank", "statement"]
}
```
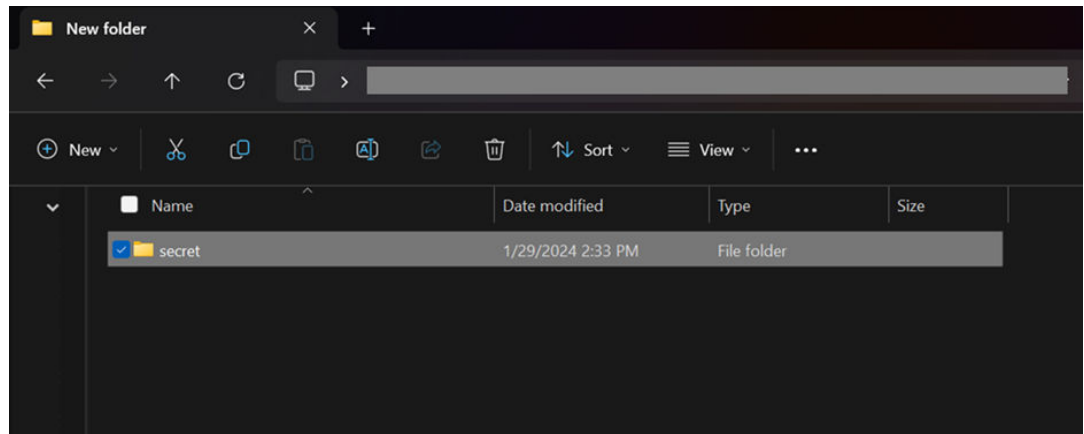
2.4 OCI Credentials and Configuration Setup

This topic provides the systematic instructions on the OCI Credentials and Configuration Setup.
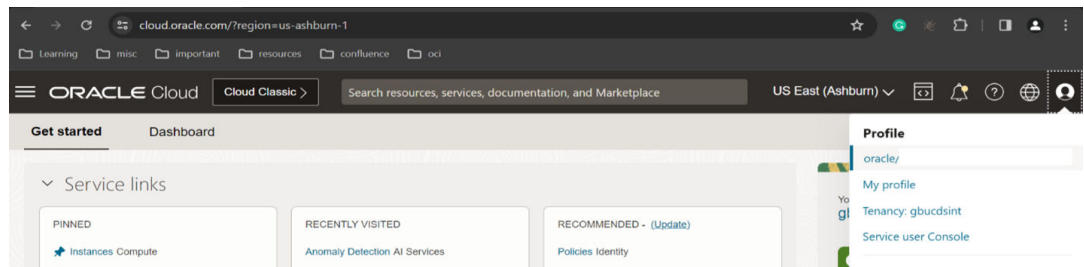
> **Note:**
>
> Subscription for the **Document Understanding** services in OCI is required. Oracle Cloud Identifier (OCID) of the compartment is required to access the Document Understanding services.
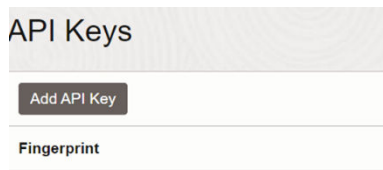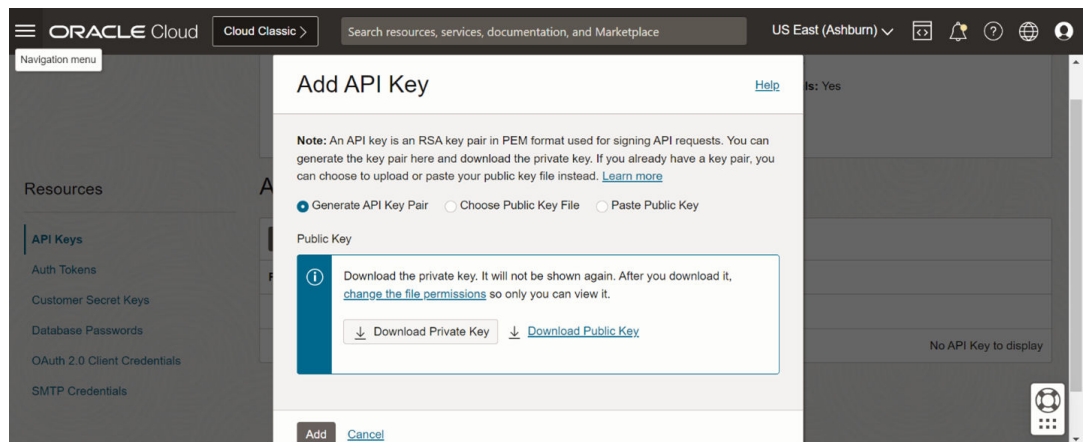
1. Create a folder **secret**.

2. Login to OCI with your credentials.

3. Open the **Profile** menu and click **My profile**.
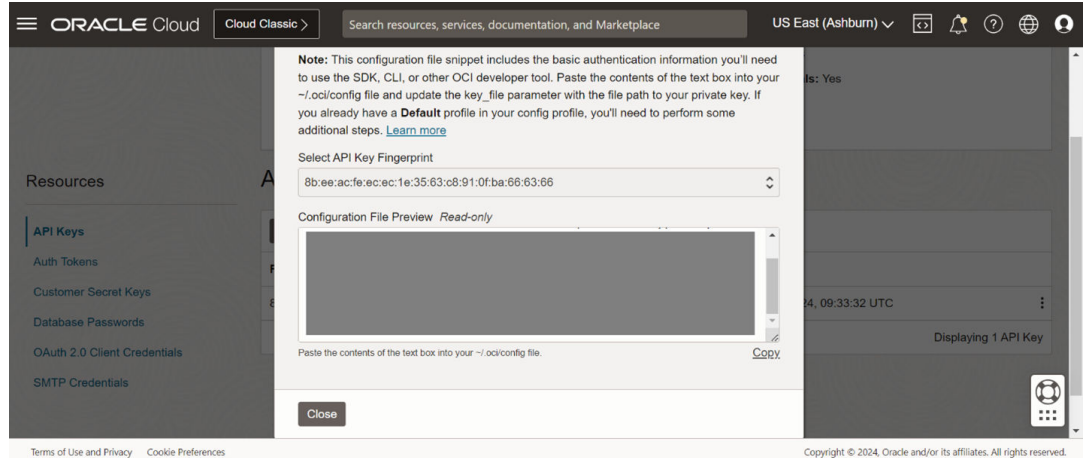

4. In the **Resources** section at the bottom left, click **API Keys**.
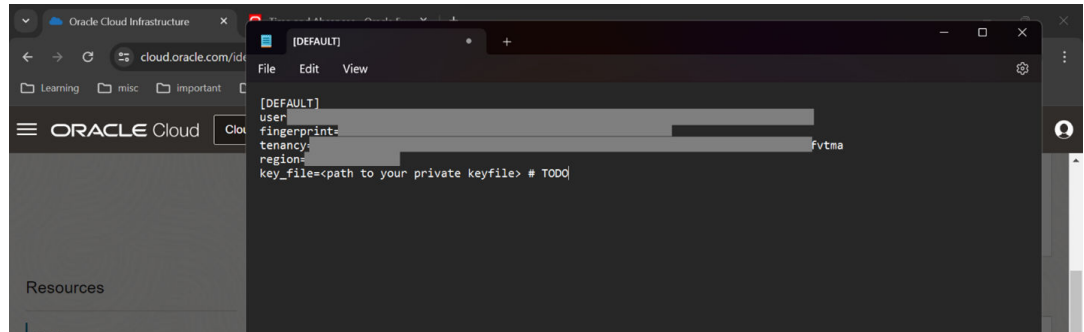

5. Click **Add API Key** at the top left of the **API Keys** list. The **Add API Key** dialog displays.
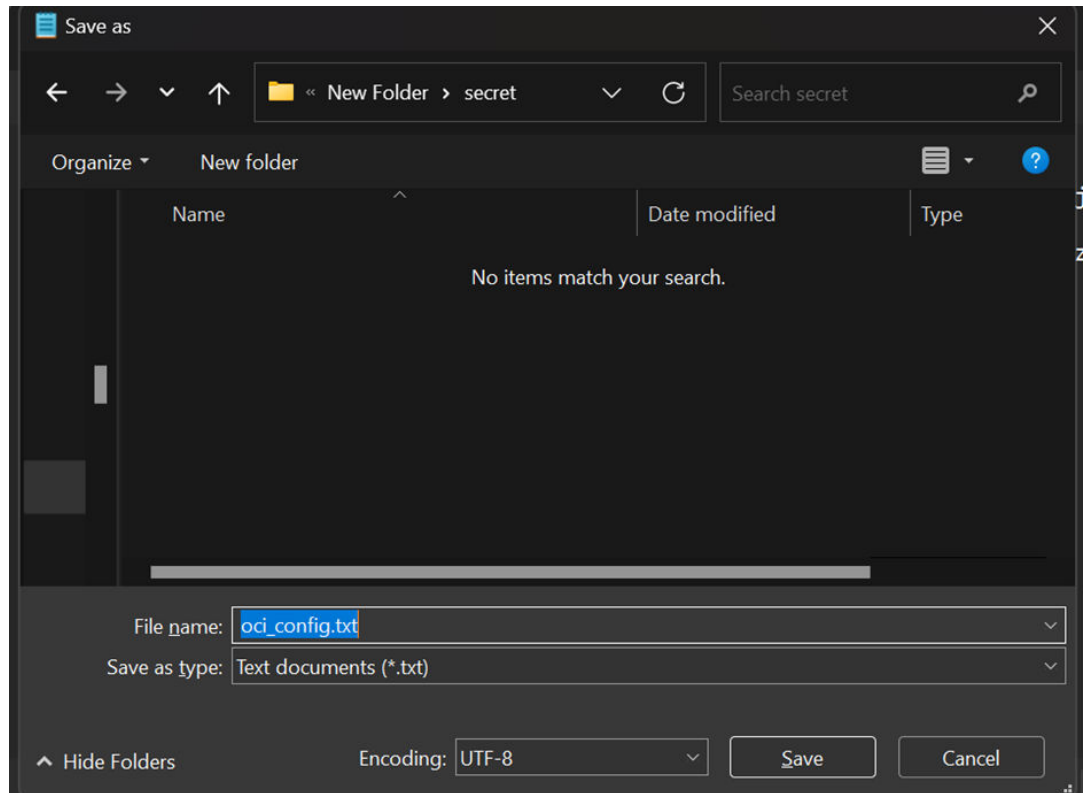
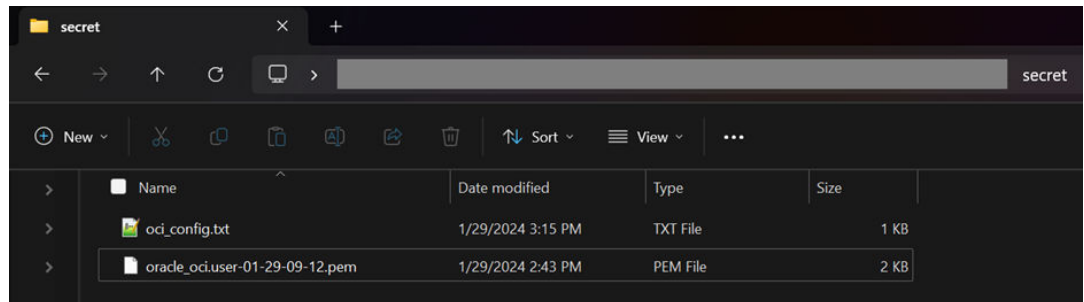6. Click **Download Private Key**, the file gets downloaded.

7. Click **Add**. A Pop-up window **Configuration File Preview** is displayed. Copy the content of file in a text file.
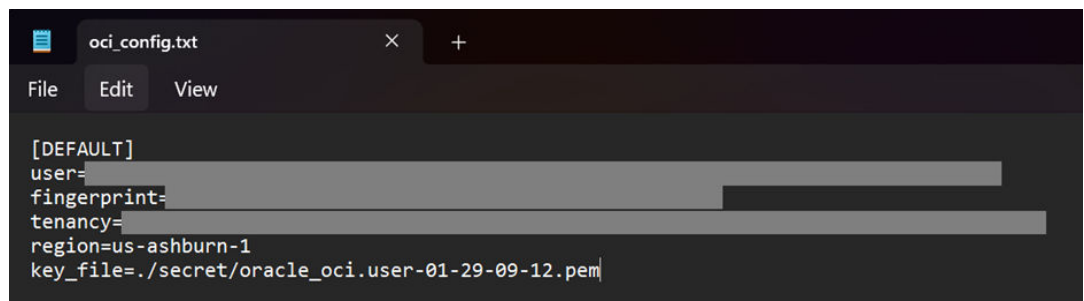

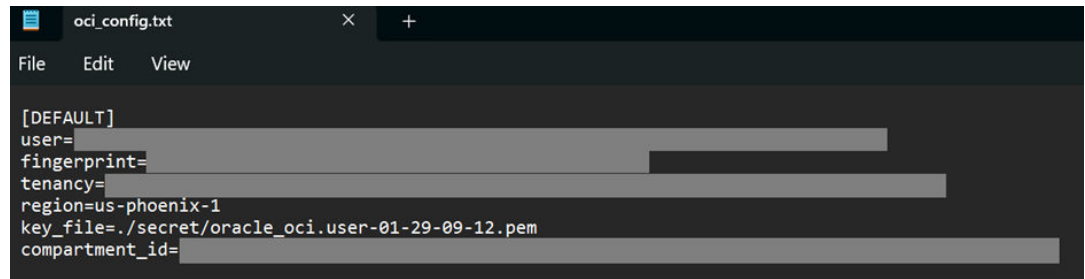8. Save the file in the secret folder.

9. Copy the Private Key file downloaded in the secret folder created.

10. Edit **oci_config.txt** file. Change the key file path to the path of the private file in the secret folder. For example: key_file=./secret/oracle_oci.user-01-29-09-12.pem.

11. In the **oci_config.txt** file, add a parameter name **compartment_id**. Specify the OCID of the compartment you will use for Document Uunderstanding Services.

```
[DEFAULT]
user=
fingerprint=
tenancy=
region=us-phoenix-1
key_file=./secret/oracle_oci.user-01-29-09-12.pem
compartment_id=
```

12. Save the file **oci_config.txt**.

13. Move the **logging-config.json**, **system-config.json** and **application-config.json** to the current working directory.

14. Make sure the below folder structure is followed:

   • root_dir

      – secret

      – Config.ini

      – system-config.json

      – application-config.json

      – logging-config.json

3
Starting The Application

This topic describes the information to start the application process.

Make sure the wheel package and the dependencies are installed, and OCI Credentials and configuration setup is complete.

1. Run the genai_doc_analyzer server using the below-mentioned command:

   ```
   python -m genai_doc_analyzer
   ```

2. By default the application runs on port 7777. You can change the port by passing **–p** argument.

   For example: `python -m genai_doc_analyzer -p 5000`

3. To run the service in the background, use the command below.

   ```
   nohup python -m genai_doc_analyzer > nohup.txt
   ```

   > **Note:**
   >
   > After the execution of the above command, all the execution logs will be added to nohup.txt which is a text file. Now you may close the terminal and the application will keep running on port, unless stopped explicitly.

4. By default the application starts on http. You can change the protocol by passing the **-s** argument.

   ```
   python -m genai_doc_analyzer -p 5000 -s https
   python -m genai_doc_analyzer -p 5000 -s
   ```

   This starts the application on both http and https protocols on ports 5000 and 5001 respectively.

   > **Note:**
   >
   > To run the service on https port, set the **SSL_CERTIFICATE and SSL_KEY** paths as mentioned in the Configuration Update (page 2-3) section.

5. To terminate or kill the application, use the netstat command to find the **process_id** using the port on which the application is running. Then use the kill command with the **process_id** as shown below to terminate the application.

   ```
   netstat -nlp | grep 7777
   kill -9 <process_id>
   ```

Index