

The screenshot shows a code editor interface with the following details:

- File Explorer (Left):** Shows the project structure under "OPEN EDITORS".
 - cloud_proc.py (active)
 - README.md
 - ! docing-stack.yaml
 - infra
 - lambda
 - Dockerfile
 - handler.py
 - requirements.txt
 - DOCLING-SERVERLESS
 - lambda-trust.json
 - response.json
 - sample.docx
 - sample.error.txt
 - sample.md
 - sample.pdf
 - sample2.error.txt
 - sample10.error.txt
 - sample14.error.txt
 - terminal_history.txt
- Editor Area (Center):** Displays the content of `cloud_proc.py`. The code defines a function `upload_and_wait` which takes `s3`, `bucket`, `filepath`, `poll_sec`, and `timeout_sec` as parameters. It uses `boto3` to upload the file and `botocore.exceptions.ClientError` to handle errors. It also includes logic to determine output keys based on the input file name.

```
2 import os
3 import sys
4 import time
5 import argparse
6 from concurrent.futures import ThreadPoolExecutor, as_completed
7
8 import boto3
9 from botocore.exceptions import ClientError
10
11 SUPPORTED = (".pdf", ".docx")
12
13
14 def to_output_keys(filename: str):
15     base = os.path.basename(filename)
16     root, _ = os.path.splitext(base)
17     return f"output/{root}.md", f"output/{root}.error.txt"
18
19
20 def upload_and_wait(s3, bucket: str, filepath: str, poll_sec: float, timeout_sec: int):
21     base = os.path.basename(filepath)
22     in_key = f"input/{base}"
23     out_key, err_key = to_output_keys(filepath)
```
- Terminal (Bottom):** Shows a PowerShell terminal window with the command `PS C:\Users\obini\Downloads\docling-serverless> []`.
- Status Bar (Bottom):** Shows the current file is `cloud_proc.py`, the line number is 23, the column number is 48, and the encoding is UTF-8.