

```

import io
import json
import os
import oci
from datetime import datetime

NAMESPACE = os.environ.get("NAMESPACE")
BUCKET_NAME = os.environ.get("BUCKET_NAME")

def handler(ctx, data: io.BytesIO = None):
    try:
        payload = json.loads(data.getvalue())
        eventTime = payload.get("eventTime", "")
        eventType = payload.get("eventType", "")
        data_section = payload.get("data", {})
        resourceName = data_section.get("resourceName", "")
        add_details = data_section.get("additionalDetails",
    {})
        requestType = add_details.get("requestType", "")
        status = add_details.get("status", "")
        # ---- Build human-readable event summary ----
        text_summary = (
            "Oracle Fusion Analytics Pipeline Event
Summary\n"
            "-----
-\n"
            f"Event Time: {eventTime}\n"
            f"Event Type: {eventType}\n"
            f"Resource Name: {resourceName}\n"
            f"Request Type: {requestType}\n"
            f>Status: {status}\n"
            "\nThis notification was automatically generated
by Oracle Fusion Analytics Pipeline.\n"
        )

        # ---- Alternatively, build a CSV-style summary ----
        csv_summary = (

```

```

"eventTime,eventType,resourceName,requestType,status,completionTime\n"

f"{eventTime},{eventType},{resourceName},{requestType},{status}\n"
    )
    # ---- Upload to OCI Object Storage ----
    signer =
oci.auth.signers.get_resource_principals_signer()
    object_storage =
oci.object_storage.ObjectStorageClient(config={},
signer=signer)
    timestamp =
datetime.utcnow().strftime('%Y%m%dT%H%M%SZ')
    object_name_text = f"pipeline_event_{timestamp}.txt"
    object_name_csv = f"pipeline_event_{timestamp}.csv"
    # Upload text summary
    object_storage.put_object(
        NAMESPACE,
        BUCKET_NAME,
        object_name_text,
        text_summary.encode('utf-8')
    )
    # Upload CSV summary
    object_storage.put_object(
        NAMESPACE,
        BUCKET_NAME,
        object_name_csv,
        csv_summary.encode('utf-8')
    )
    print(f"Event summaries uploaded as
{object_name_text} and {object_name_csv}")
    except Exception as ex:
        print('ERROR: Failed to process or upload event
data:', ex, flush=True)
        raise

```