National College of Ireland

Configuration Manual

MSc Research Project
Msc in Cloud Computing

Sanjith chokalingam Pillai
subramonia pillai
Student ID: 23194383

School of Computing
National College of Ireland

Supervisor: Yasantha Samarawickram

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sanjith Chkalingam Pillai Subramonia Pilliai
Student ID:	23194383
Programmee:	Msc in Cloud Computing **Year:** 2024-2025
Module:	Research Project
Lecturer:	Yasantha Samarawickarma
Submission Due Date:	15.09.2025
Project Title:	A Serverless Pipeline Framework with Dynamic Schema Adaptation for Enhanced CSV Processing in AWS

Word Count: **300** **Page Count: 4**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Sanjith Chokalingam Pillai Subramonia Pillai

Date: 11.08.2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	✓

Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	✓
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	✓

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Forename Surname
Student ID:

1 Environment Requirements

1.1 System Requirements

- OS: Windows/Linux
- RAM: Min 16GB
- Storage: SSD Preferred with no less than 256GB
- Network: Stable Internet for AWS CLI access

1.2 AWS Services

- AWS Lambda: Serverless compute platform
- AWS Step Functions: Workflow orchestration
- Amazon S3: Object storage for CSV files
- Amazon ECR: Container registry for Lambda images
- CloudWatch: Monitoring and logging

1.3 Development Environment

- Python: Version 3.9 or higher
- Docker: Version 20.x or higher (with DOCKER_BUILDKIT=0 for AWS Academy)
- AWS CLI with credentials authorised.

2 Package Requirements

```
# Core packages (requirements.txt)
pandas>=1.5.3
numpy>=1.21.0
scipy>=1.9.0
boto3>=1.26.0
pytest>=7.0.0
python-dotenv>=0.19.0

# Development packages
black>=22.0.0
flake8>=4.0.0
mypy>=0.950
```

3 Installation Procedure

```
cd serverless-csv-pipeline

# Create Python virtual environment
python3 -m venv venv

# Activate virtual environment
# Windows:
venv\Scripts\activate
# Linux/macOS:
source venv/bin/activate

# Install Python dependencies
pip install -r requirements.txt
```

4 AWS Configuration

```
# Configure AWS credentials (AWS Academy)
aws configure
# AWS Access Key ID: [From AWS Academy]
# AWS Secret Access Key: [From AWS Academy]
# Default region: us-east-1
# Default output format: json

# Verify configuration
aws sts get-caller-identity
```

5 Deployment

```
# Create Step Functions definition
cat > step-functions-pipeline.json << 'EOF'
{
  "Comment": "Serverless CSV Processing Pipeline",
  "StartAt": "DataIngestion",
  "States": {
    "DataIngestion": {
    "Type": "Task",
      "Resource": "arn:aws:states:::lambda:invoke",
      "Parameters": {
        "FunctionName": "csv-data-ingestion",
        "Payload.$": "$"
      },
      "ResultPath": "$.ingestion_result",
      "Next": "SchemaDetection"
    },
    "SchemaDetection": {
      "Type": "Task",
      "Resource": "arn:aws:states:::lambda:invoke",
```

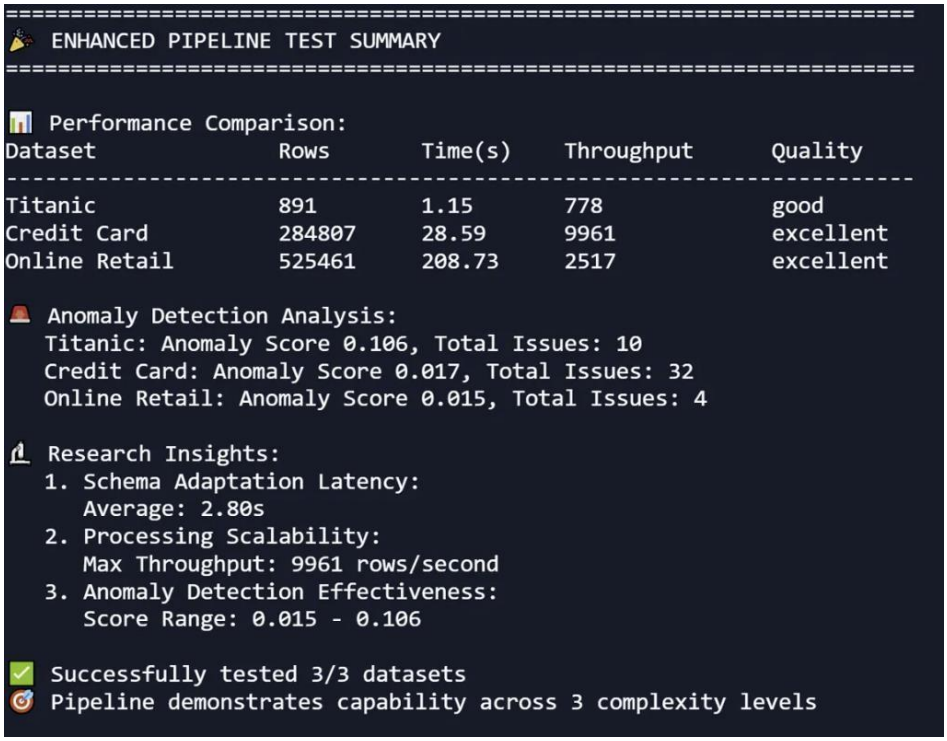
```
      "Parameters": {
        "FunctionName": "csv-schema-detection",
        "Payload.$": "$.ingestion_result.Payload"
      },
      "ResultPath": "$.schema_result",
      "Next": "AnomalyDetection"
    },
    "AnomalyDetection": {
      "Type": "Task",
      "Resource": "arn:aws:states:::lambda:invoke",
      "Parameters": {
        "FunctionName": "csv-anomaly-detection",
        "Payload": {
          "bucket.$": "$.schema_result.Payload.bucket",
          "key.$": "$.schema_result.Payload.key",
          "processing_id.$": "$.schema_result.Payload.processing_id",
          "schema.$": "$.schema_result.Payload.schema"
        }
      },
      "End": true
    }
  }
}
EOF

# Deploy state machine
aws stepfunctions create-state-machine \
  --name "ServerlessCSVPipeline" \
  --definition file://step-functions-pipeline.json \
  --role-arn "arn:aws:iam::ACCOUNT_ID:role/LabRole"
```

6 Test

python tests/test_pipeline_v2.py

7 Results



```
======================================================================
🎉 ENHANCED PIPELINE TEST SUMMARY
======================================================================

📊 Performance Comparison:
Dataset              Rows       Time(s)    Throughput      Quality
----------------------------------------------------------------------
Titanic              891        1.15       778             good
Credit Card          284807     28.59      9961            excellent
Online Retail        525461     208.73     2517            excellent

🚨 Anomaly Detection Analysis:
   Titanic: Anomaly Score 0.106, Total Issues: 10
   Credit Card: Anomaly Score 0.017, Total Issues: 32
   Online Retail: Anomaly Score 0.015, Total Issues: 4

🔬 Research Insights:
   1. Schema Adaptation Latency:
      Average: 2.80s
   2. Processing Scalability:
      Max Throughput: 9961 rows/second
   3. Anomaly Detection Effectiveness:
      Score Range: 0.015 - 0.106

✅ Successfully tested 3/3 datasets
🎯 Pipeline demonstrates capability across 3 complexity levels
```