National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Pranal Kale
Student ID: 23312599

School of Computing
National College of Ireland

Supervisor: Yasantha Samarawickrama

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Pranal Kale
Student ID:	23312599
Programme:	Cloud Computing
Year:	2025
Module:	MSc Research Project
Supervisor:	Yasantha Samarawickrama
Submission Due Date:	28/01/2026
Project Title:	Configuration Manual
Word Count:	800
Page Count:	6

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:	
Date:	28th January 2026

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

Pranal Kale
23312599

28th January 2026

Contents

1 Introduction

This document provides an overview of the various software artifacts used in research related to "Dynamic Resource Scaling Utilizing Predictive Analytics" as well as guidelines for how to install, configure, and execute them.

The Dynamic Scaling System is implemented using both local (Python) predictive models and remote (cloud-based) infrastructure provided through Amazon Web Services (AWS). Its workflow includes: provisioning an AWS infrastructure; processing historic work loads for prediction; developing machine-learning (including Random Forest, Linear Regression, and Long Short-Term Memory Networks) prediction models; running simulators to test AWS configuration and automatic scaling functionality.

Following the procedures described in this document will allow others to repeat the processes that were performed in order to produce these experiments and validate the auto-scaling function in a controlled test setting. Acknowledge that AWS is necessary in order to complete the tasks described in the document using AWS environments and a local machine(s) capable of running a Python application.

2 System Prerequisites

Before deploying the predictive scaling system, verification must be performed to ensure the host machine meets the following requirements:

- **Operating System:** Windows 10/11, macOS, or Linux (Ubuntu 20.04+ recommended).
- **Python Version:** Python 3.11.5 or higher.
- **Network:** Active internet connection to communicate with AWS Endpoints.
- **AWS Account:** An active AWS account with Administrator access to create IAM roles, EC2 instances, and CloudWatch alarms.

3 Project Directory Structure

The operational integrity of the system relies on a specific file structure. Ensure the project files are organized as follows:

```
project_root/
|-- config.py                          # Global configuration and credentials
|-- main.py                            # Central orchestration script
|-- requirements.txt                   # Python dependencies
|-- phase1_aws_infrastructure_setup.py # Infrastructure provisioning
|-- phase2_data_preprocessing.py       # ETL pipeline
|-- phase3_predictive_model.py         # Model training (RF, LR, LSTM)
|-- phase4_dynamic_scaling.py          # Scaling logic and simulation
|-- phase5_performance_evaluation.py   # Reporting and visualization
|-- cloud-key-pair.pem                 # EC2 Private Key
|-- dataset/
|   `-- cloud_workload_dataset.csv     # Historical workload data
```

4 Installation

4.1 Virtual Environment Setup

To maintain dependency isolation, the use of a virtual environment is mandated.

Listing 1: Creating and Activating Virtual Environment

```
# Windows
python -m venv venv
.\venv\Scripts\activate

# Linux/macOS
python3 -m venv venv
source venv/bin/activate
```

4.2 Installation Dependency

The required Python libraries must , must be installed using the provided requirements file.

```
pip install -r requirements.txt
```

Required libraries:

- `boto3`: AWS SDK for Python.
- `pandas numpy`: data processing.
- `scikit-learn`: Machine learning algorithms.
- `tensorflow`: Deep , Deep learning (for LSTM).
- `matplotlib seaborn`: visualization.

5 Configuration

The system is controlled via the `config.py` file. This file must be populated with valid AWS credentials and region-specific parameters.

5.1 AWS Credentials

Warning: /The `config.py` file contains sensitive encryption KEYS and should not be stored in public repositories.

Listing 2: config.py Configuration

```
"""
Configuration file for AWS credentials and project settings
"""

# AWS Configuration
AWS_REGION = 'eu-north-1'  # Stockholm
```

```
AWS_ACCOUNT_ID = '066833532098'
# Replace with your actual IAM User Credentials
AWS_ACCESS_KEY_ID = 'AKIAQ7D4WYTBKUAD42X6'
AWS_SECRET_ACCESS_KEY = 'iI4m1s+4MkLcXdgyTSkpZ2FKEvS4p5oUiwMorMr4'

# EC2 Configuration
KEY_PAIR_NAME = 'cloud-key-pair'
# Ensure this path is absolute and uses forward slashes
KEY_PAIR_PATH = 'f:/pranal_thesis/cloud-key-pair.pem'
AMI_ID = 'ami-07fb0a5bf9ae299a4'  # Amazon Linux 2023 AMI
INSTANCE_TYPE = 't3.small'

# S3 Configuration
S3_BUCKET_NAME = 'cloud-scaling-data-pranal'

# Project Configuration
PROJECT_NAME = 'dynamic-resource-scaling'
DATASET_PATH = 'f:/pranal_thesis/dataset/cloud_workload_dataset.csv'

# Scaling Thresholds
CPU_SCALE_UP_THRESHOLD = 70
CPU_SCALE_DOWN_THRESHOLD = 30
```

6 Execution Guide

The project uses a central orchestration script (`main.py`) to manage the execution of all phases.

6.1 Running the Orchestrator

Execute the bellow command in your terminal:

```
python main.py
```

6.2 Operation Menu

On execution, the below menu will appear:

```
----------------------------------------------------------------------
SELECT PHASE TO EXECUTE:
----------------------------------------------------------------------
  0. Run Complete Pipeline (All Phases)
  1. Phase 1: AWS Infrastructure Setup
  2. Phase 2: Data Preprocessing and Analysis
  3. Phase 3: Predictive Model Training
  4. Phase 4: Dynamic Scaling Implementation
  5. Phase 5: Performance Evaluation
  6. Install Dependencies
```

```
  7. Exit
--------------------------------------------------------------------
```

- **Option 0 (Recommended):** Runs the full pipeline together. This ensures data is processed, models are trained, and simulation is run in the correct order.
- **Option 1:** Provisions the S3 Bucket, IAM Roles, Launch Templates, and Auto Scaling Group. *Note: This is idempotent; running it twice will not create duplicate resources.*
- **Option 4:** Runs the simulation. It loads the trained models from `phase3` and generates traffic patterns to test scaling logic.

7 Output Artifacts

After successful execution, the following directories and files will be generated:

1. **/models**: Contains serialized ML models (`random_forest_cpu.pkl`, `linear_regression.pkl`) and scalers.
2. **/analysis_output**: Contains EDA graphs (Correlation Matrix, Hourly Usage Patterns).
3. **/evaluation_output**: Contains the final report images (Cost Analysis, Model Comparison) and JSON reports.
4. **simulation_results.json**: A log of every decision made during the 30-minute simulation.
5. **scaling_events.log**: A text-based log of system activities.

8 Troubleshooting

Error	Cause	Solution
`ClientError: Invalid AMI ID`	The AMI ID in `config.py` is not available in the selected region (`eu-north-1`).	Verify the AMI ID matches the region in the AWS Console.
`FileNotFoundError: dataset.csv`	The path in `DATASET_PATH` is incorrect.	Ensure the path uses forward slashes and points to the correct location.
`Boto3 Credentials Error`	The Access Key or Secret Key is invalid.	Regenerate keys in IAM and update `config.py`.

9 Conclusion

By following the procedures outlined in this manual, organizations will have the ability to effectively implement and run a predictive scaling solution. By completing the configuration steps outlined here, organizations will have created all required AWS infrastructure

(EC2 instances, Auto Scaling Groups, and CloudWatch alarms) to support this solution in a reproducible manner. When this pipeline is executed, machine learning models will be trained using historical data(s) and this will drive all decisions regarding the organization's cloud infrastructure. Once all steps are completed successfully, comprehensive evaluation metrics and visualization artifacts will be produced, demonstrating that the developed codebase supports the management of cloud resources in real time.