National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Libin Tom Kuriyakose Kadavil
Student ID: 21204420

School of Computing
National College of Ireland

Supervisor: Dr. Diego Lugones

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Libin Tom Kuriyakose Kadavil
Student ID:	21204420
Programme:	Cloud Computing
Year:	2022
Module:	MSc Research Project
Supervisor:	Dr. Diego Lugones
Submission Due Date:	14/08/2023
Project Title:	Configuration Manual
Word Count:	570
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:	
Date:	18th September 2023

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

Libin Tom Kuriyakose Kadavil
21204420

1 Data Analysis and ARIMA Modeling

Data analysis and the Time series forecasting has been performed in the google Colaboratory (interactive notebooks allows to combine executable code and rich text in a single document) it is a cloud platform solution provided by Google inc. It is popularly used in machine learning areas for data analysis model training, testing and evaluation before deploying the final model in the infrastructure.

1.1 Import Data and Convert to Data Frames

1.1.1 Import Libraries

- pandas
- json
- numpy
- matplotlib.pyplot

1.1.2 Import and Convert Data

- Import by providing the file dataset path
- Set the timestamp to pandas date time format to a variable

```
[pd.to_datetime(ts, unit='ms') for ts in dataset_name["dict"][0]]
```

- Select and store the utilization in another variable from the dataset

```
dataset_name["dict"][1]]
```

- Convert the data to pandas data frame using pandas library

```
pd.DataFrame({'timestamp': timestamp, 'utilization': utilizations})
```

- Set the timestamp as dataframe index for further analysis
- Resample the data using resample() function group the dataset to daily max value

```
data.resample('D').max()
```

- Plot the daily utilization data using the pyplot

1.2 Stationarity Checking

1.2.1 Import Libraries

- statsmodels.api
- statsmodels.tsa.seasonal seasonal_decompose
- statsmodels.tsa.stattools adfuller
- matplotlib.pyplot

1.2.2 Seasonal Decomposition and Adfullar Test

- Perform the additive model seasonal decomposition with 24 * 6 on both original and daily max data
- Plot the decomposition statistic observed, trend, seasonal and residual
- Perform Adfuller test

1.2.3 Rolling Statistics

- Perform rolling statistics mean and std deviating using rolling() function of the pandas

1.3 Data Transformation

1.3.1 Log and Difference Transformation

- Perform Log Transformation using numpy log() function
- Perform Difference Transformation by subtracting log data by log's shift using numpys shift () function
- Plot the Original, Log and Diff

1.3.2 Rerun the Adfuller

- Perform the adfuller test in the tranformed data

1.4 Build ARIMA Model

1.4.1 Import Libraries

- statsmodels.tsa.arima.model ARIMA
- warnings to ignore the warnings

1.4.2 Build Model

- Fit the ARIMA model with data and order of p, d, q (0,0,7)

1.4.3 Prediction

- Predict the future utilization using the arima predict () function with start parameter as length of dataset and end with length of dataset - 1
- Print the mean of the forecast utilization
- Plot the Trained and Arima predicted utilization

2 Dynamic Deployment

Developed a python based program to automate the prediction of the CPU and deploy the Kubernetes manifesto to the Kubernets test environment by providing the dataset and other required parameters from the user. The program will predict the cpu and then replace all the required values and generates the manifesto and creates back of the manifesto in a YAML file and the deploys the manifesto to the Kubernetes environment using pythons Kubernetes client library. In the following subsection will explain how to deploy the code in the test environment.

2.1 Setup the Kubernetes Environment

- Install docker
- Install Minikube
- Start minikube to use the driver as docker daemon

```
minikube start --driver=docker
```

2.2 Execute the Python Program

- Create a python virtual environment
- Install the library's from the requirements.txt file included with the code
- Execute the python program - This program will provide outputs of the prediction and deployment

```
python3 predicted-pod-depolyment.py --ts_data prometheus-dataset-1.json
--pod_name research-app-1
--image nginx:1.15.4
--cpu_request 500m
--memory_request 500Mi
--memory_limit 1Gi
--no_replicas 2
--port_num 80
```

- predicted-pod-depolyment.py - Python program file
- –ts_data - Dataset path

- –pod_name - Desired name for the POD
- –image - Container image name and version
- –cpu_request - Minimum CPU to set for the POD
- –memory_request - Minimum meomory to set for the POD
- –memory_limit - Maximum memory to set for the POD
- –no_replicas - No. of POD replicas in the deployment
- –port_num - Application port number

2.3 Test the Deployment

- Check the POD's are deployed

```
kubectl get pods
```

- Describe the POD to check the CPU Limit

```
kubectl describe pod pod-name
```