

Configuration Manual

MSc Research Project
Cloud Computing

Kirthikesh Parthasarathy
Student ID:21195391

School of Computing
National College of Ireland

Supervisor: Shreyas Setlur Arun

National College of Ireland
Project Submission Sheet
School of Computing



Student Name:	Kirthikesh Parthasarathy
Student ID:	21195391
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Shreyas Setlur Arun
Submission Due Date:	14/12/2023
Project Title:	Configuration Manual
Word Count:	951
Page Count:	8

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:	Kirthikesh Parthasarathy
Date:	29th January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies).	<input type="checkbox"/>
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.	<input type="checkbox"/>

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

Kirthikesh Parthasarathy
21195391

1 Introduction

This configuration manual provides full instructions for the creation and execution of the research project. "An Enhanced Version of Data Classification based on Confidentiality for Cloud Security". The manuals provide a clear and organised instructions for installing all of the dependencies, libraries, and packages needed to implement the project. Then, it discusses the entire architecture from an overview, initially providing the reader with information of each of the parts and how they are connected. The system installation provides an in-depth overview of the execution flow and processes, providing users a precise understanding of the interactions between modules during operation. The book also provides troubleshooting guidance for common mistakes and difficulties that may arise during installation or execution. By following the methods provided in this in-depth manual, the reader will have the required knowledge to successfully implement the project in a different environment. It is recommended that anyone planning to operate the system carefully read this manual full.

2 Hardware and Software Requirements

2.1 Software Requirements

MacOS and Linux has been used for this research. And i have developed this project with Python 3.7 and later version can be used if its required. To download the python latest version use this url link: <https://www.python.org/downloads/>

Component	Specification
Operating System	Linux
Main Memory	4GB
Number of Cores	8 (Virtual cores)
Storage	30GB
Programming Language	Python3
Python Libraries	Numpy, Pandas, mysqlclient, Keras, TensorFlow, scikit-learn, Flask

Table 1: System Specifications and Software Stack

2.2 Hardware Requirements

Before using this software, it is necessary for users to have a basic knowledge of Amazon Cloud, Anaconda. Navigator, Jupyter Notebook, Python programming, and frequent

machine learning models. Knowing a basic knowledge in Jupyter Notebook is necessary to effectively explore the file system, install packages, and perform commands through the terminal. Users should possess a basic understanding of, including the Jupyter administration of software, users, and processes. Knowledge in the Python programming language is also required. Users must possess a strong knowledge with Python syntax, data structures, control flow, modules, and virtual environments. Also, it is essential for users have knowledge in machine learning algorithms and models training, testing, and performance evaluation.

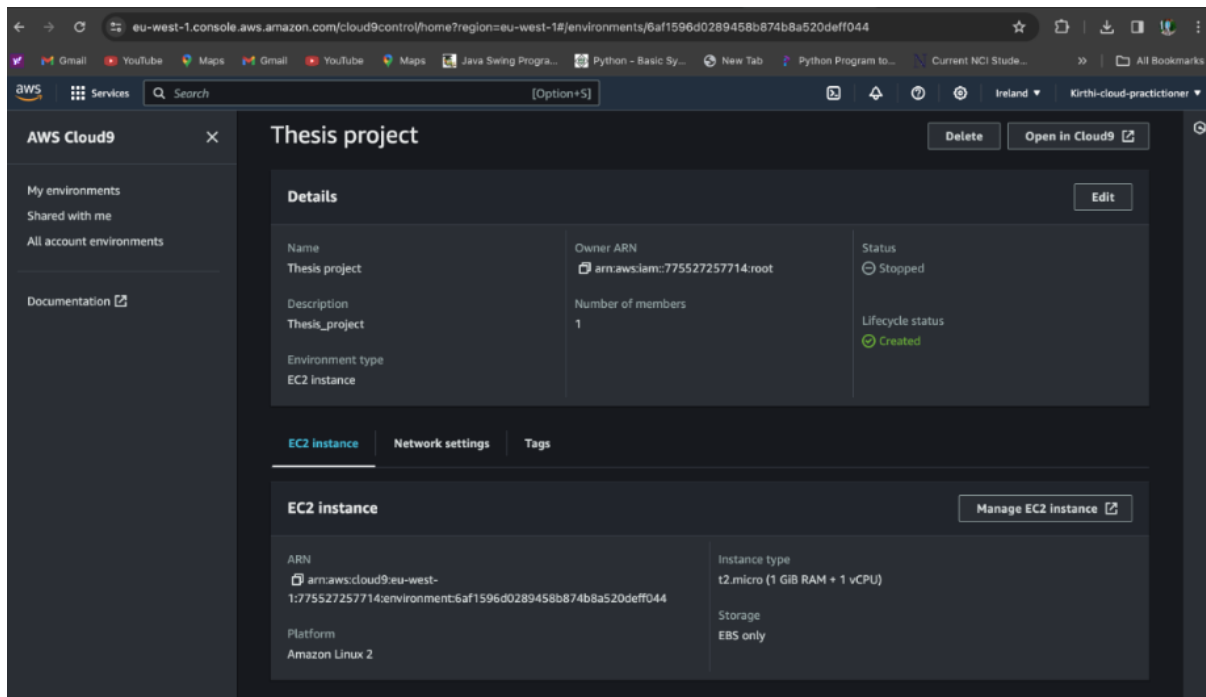


Figure 1: AWS EC2 Instance

3 Software Installation

- Install Anaconda3 using this link: <https://www.anaconda.com/download>. ?
- Install Python: Make sure you have Python installed on your computer?. You can download the latest version from the official Python website. Jupyter Notebook comes pre-installed with Anaconda distribution as well. <https://www.python.org/downloads/>

```
(myenv) ec2-user:~/environment/Final_Update_01 $ sudo yum install python3.7
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
246 packages excluded due to repository priority protections
Package python3-3.7.16-1.amzn2.0.4.x86_64 already installed and latest version
```

Figure 2: sudo yum install python3.7

- Install Jupyter: You can install Jupyter Notebook using the following command in your terminal or command prompt: `pip install jupyter`

```

Last login: Wed Dec 6 02:32:57 on ttys000
/Users/kirthikesh/anaconda3/bin/jupyter_mac.command ; exit;
(base) kirthikesh@kirthikeshs-Air ~ % /Users/kirthikesh/anaconda3/bin/jupyter_mac.command ; exit;

┌───────────┐
│             │
│             │
└───────────┘

Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions.
https://jupyter-notebook.readthedocs.io/en/latest/migrate_to_notebook7.html

Please note that updating to Notebook 7 might break some of your extensions.

[W 02:57:14.130 NotebookApp] Loading JupyterLab as a classic notebook (v6) extension.
[I 2023-12-10 02:57:14.133 LabApp] JupyterLab extension loaded from /Users/kirthikesh/anaconda3/lib/python3.11/site-packages/jupyterlab
[I 2023-12-10 02:57:14.133 LabApp] JupyterLab application directory is /Users/kirthikesh/anaconda3/share/jupyter/lab
[I 02:57:16.917 NotebookApp] Serving notebooks from local directory: /Users/kirthikesh
[I 02:57:16.917 NotebookApp] Jupyter Notebook 6.5.4 is running at:
[I 02:57:16.917 NotebookApp] http://localhost:8888/?token=0ef1a648eae1c0ba5f9781c1021722dc3cd60f7c5eb20841
[I 02:57:16.917 NotebookApp] or http://127.0.0.1:8888/?token=0ef1a648eae1c0ba5f9781c1021722dc3cd60f7c5eb20841
[I 02:57:16.917 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

[C 02:57:16.925 NotebookApp]

To access the notebook, open this file in a browser:
file:///Users/kirthikesh/Library/Jupyter/runtime/nbserver-92112-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=0ef1a648eae1c0ba5f9781c1021722dc3cd60f7c5eb20841
or http://127.0.0.1:8888/?token=0ef1a648eae1c0ba5f9781c1021722dc3cd60f7c5eb20841
0.00s - Debugger warning: It seems that frozen modules are being used, which may
0.00s - make the debugger miss breakpoints. Please pass -xfrozen_modules=off
0.00s - to python to disable frozen modules
0.00s - Note: Debugging will proceed. Set PYDEVD_DISABLE_FILE_VALIDATION=1 to disable this validation.

```

Figure 3: pip install jupyter

- Open Terminal/Command Prompt: Once you have Jupyter installed, open your terminal (Linux/Mac) or command prompt (Windows).
- Navigate to the Desired Directory: Use the cd command to navigate to the directory where you want to create or access your Jupyter Notebook files. To access the notebook, open this file in a browser:
file:///Users/kirthikesh/Library/Jupyter/runtime/nbserver-92112-open.html.
- Start Jupyter Notebook: In the terminal/command prompt, enter the following command:jupyter notebook
- This will start the Jupyter Notebook server and open a new tab or window in your default web browser.

```

In [1]: 1 import os
        2
        3 import csv
        4
        5 def classify_data(input_text):
        6     # Define a dictionary of attribute names and their corresponding keywords
        7     attribute_keywords = {
        8         "Personal Identifiable Information (PII)": ["names", "social security numbers", "birthdates", "addresses",
        9             "phone numbers", "email addresses"],
       10         "Financial Data": ["credit card numbers", "bank account numbers", "financial transaction records", "profits"],
       11         "Healthcare Information (PHI)": ["medical records", "health insurance information", "prescription information"],
       12         "Intellectual Property": ["patents", "trade secrets", "copyrighted materials", "research and development data"],
       13         "Customer Data": ["customer contact information", "purchase history", "customer preferences"],
       14         "Employee Data": ["payroll information", "human resources records", "employee contracts"],
       15         "Legal Documents": ["contracts", "legal correspondence", "court documents"],
       16         "Sensitive Business Data": ["business plans", "strategic documents", "financial reports"],
       17         "Logins and Passwords": ["usernames", "passwords", "access credentials"],
       18         "Confidential Communications": ["emails", "instant messages", "internal memos"],
       19         "Location Data": ["GPS coordinates", "location history"],
       20         "Biometric Data": ["fingerprints", "retina scans", "facial recognition data"],
       21         "Data Access and Authorization Information": ["access logs", "user permissions", "encryption keys"],
       22         "Third-party Data": ["data shared with third-party vendors", "partner data"],
       23         "Social Media Data": ["social media account information", "social media posts", "AdharNumber", "ID card Number"],
       24         "Government-issued Identifiers": ["passport numbers", "driver's license numbers"],
       25         "Video and Audio Recordings": ["security camera footage", "voice recordings"],
       26         "Research Data": ["scientific research findings", "experimental data"],
       27         "Mailing Lists": ["subscriber lists", "newsletter recipients"],
       28         "Customer Feedback and Surveys": ["feedback forms", "survey responses"],
       29         "Backup and Recovery Data": ["backup copies of sensitive data"],
       30         "Operational Data": ["configuration data", "network diagrams"],
       31         "Vendor and Supplier Data": ["supplier contracts", "vendor contact information"],
       32         "Metadata": ["timestamps", "file histories"],
       33         "User Profiles": ["user profiles"],
       34         "Government Data": ["Cabinet Office orders", "Freedom of Information Act", "Information commissioner", "New rule"],
       35         "Communication Data": ["E-mails", "Deletion of e-mails", "E-mail regime"],

```

Figure 4: Jupyter Notebook

4 Algorithm Implementation

4.1 DataSet Collection

During this phase, the data was categorized into two main types: confidential and non-confidential. To achieve this, key attributes were utilized for distinguishing confidential data. And the data set has been taken from kaggle and it is publicly accessible.

The textual data was collected from the Kaggle datasets which can be accessed using the following link:

[https://www.kaggle.com/datasets/jensenbaxter/10dataset-text-document-classification-select=business.?](https://www.kaggle.com/datasets/jensenbaxter/10dataset-text-document-classification-select=business.)

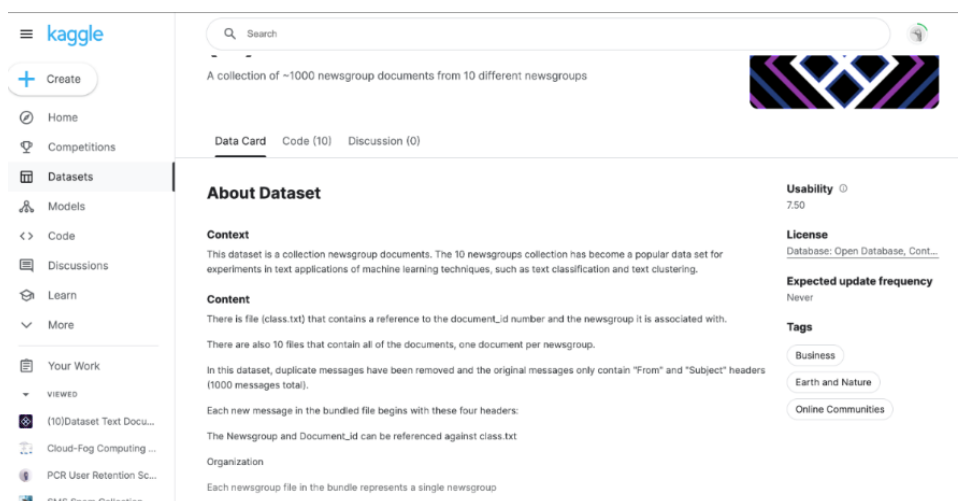


Figure 5: Kaggle datasets

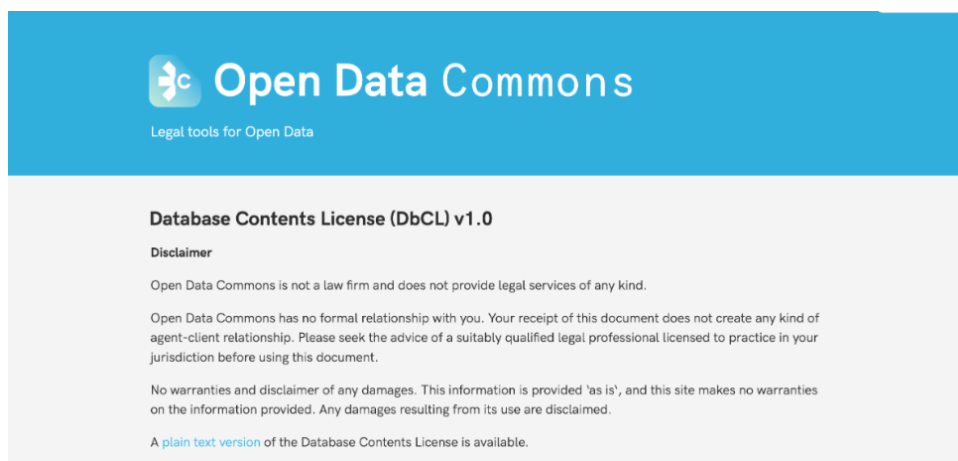


Figure 6: Proof: Database Contents Licence (DbCL) v1.0

4.2 CNN-LSTM with Random Forest Algorithm

The model is defined using TensorFlow's Keras API. It starts with a 1D convolutional layer (Conv1D) with 256 filters, a kernel size of 5, and a rectified linear unit (ReLU)

activation function. This layer is designed for causal padding, meaning it considers only the past values in the input sequence. Followed by two LSTM layers: one with 128 units and another with 64 units. Both use the hyperbolic tangent (tanh) activation function, and the first LSTM layer returns sequences. Subsequently, there are three dense layers with 32, 16, and 8 units, respectively, using ReLU activation. The last layer is a lambda layer that scales the output by a factor of 100. This scaling compensates for the fact that the LSTM's tanh activation function outputs values between -1 and 1. After applying the CNN-LSTM architecture, we obtain a set of eight features from the input data.

```

241 feature_extraction_model = tf.keras.models.Sequential([tf.keras.layers.Conv1D(filters=256, kernel_size=5,
242 strides=1, padding="causal",
243 activation="relu",
244 input_shape=[100,1]),
245 tf.keras.layers.LSTM(128, activation="tanh", return_sequences=True),
246 tf.keras.layers.LSTM(64, activation="tanh", return_sequences=False),
247 tf.keras.layers.Dense(32, activation="relu"),
248 tf.keras.layers.Dense(16, activation="relu"),
249 tf.keras.layers.Dense(8),
250 tf.keras.layers.Lambda(lambda x: x * 100) # LSTM's tanh activation returns between -1 and 1. Scaling output to same range of values helps
251 ])
252
253 # Note: to turn this into a classification task, just add a sigmoid function after the last Dense layer and remove Lambda layer.
254
255 #optimizer = tf.keras.optimizers.Adam(learning_rate=0.001, name='Adam')
256 feature_extraction_model.compile(loss='mse', optimizer='adam', metrics=tf.keras.metrics.RootMeanSquaredError(name='root_mean_squared_error'))
257
258
259
260 input_ED_Features = feature_extraction_model.predict(input_X_padded_array)
261 #clf = joblib.load('RandomForest_model_weights.pkl')
262

```

Figure 7: Code Snippet for TensorFlow's Keras API

The chosen method for secure data transfer is through the use of the Speck lightweight encryption algorithm.

```

input_ED_Features = feature_extraction_model.predict(input_X_padded_array)
#clf = joblib.load('RandomForest_model_weights.pkl')

with open('RandomForest_model_weights.sav', 'rb') as file:
    clf = pickle.load(file)
scaler=StandardScaler()
input_ED_Features_std=scaler.fit_transform(input_ED_Features)
input_y_pred=clf.predict(input_ED_Features_std)
print('mmmmmm',input_y_pred)
#key = 0x123456789ABCDEF00FEDCBA987654321
key=request.form['file_key']
my_speck = SpeckCipher(int(key))

```

Figure 8: Code Snippet for Random Forest Algorithm Implementation

After completing model development, a secure data transfer technique is implemented. The entire framework is then designed into a user interface (UI). Within the UI, users can register under the Container as a Service (CaaS) platform. They are granted the ability to store and download their own data securely.

The AUG_ConvolLSTM-RF trained model weights are used to categorise data into confidential and non-confidential segments. Notably, encryption is selectively applied only to confidential data to ensure secure storage and transfer.

```

276 with open(test_file_name+"_confidential_Encrypted.txt", "w") as confidential_file, open(test_file_name+"_nonconfidential.txt", "w") as nonconf
277     for index, (label, line) in enumerate(zip(input_y_pred, original_lines)):
278
279
280     # Check the label and write the line to the appropriate file
281     if label == 1:
282
283         long_plaintext = str(index)+": "+line
284
285         plaintext_blocks = [long_plaintext[i:i + 8] for i in range(0, len(long_plaintext), 8)]
286
287         ciphertext_blocks = []
288
289         # Encrypt each block
290         for block in plaintext_blocks:
291             # Convert the block to a 64-bit integer
292             plaintext = int.from_bytes(block.encode(), byteorder='big')
293
294             # Encrypt the block
295             ciphertext = my_speck.encrypt(plaintext)
296             # Append the ciphertext block to the list
297             print(ciphertext)
298             ciphertext_blocks.append(ciphertext)
299
300         confidential_file.write(f"{ciphertext_blocks}\n")
301     else:
302         nonconfidential_file.write(f"{index}: {line}\n")
303
304
305
306
307
308

```

Figure 9: Code snippet for Encryption and Decryption Using Speck

The code snippet of library packages that i have installed in this project

```

ec2-user~/environment $ pip install flask-mysqldb
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: flask-mysqldb in /home/ec2-user/.local/lib/python3.8/site-packages (2.0.0)
Requirement already satisfied: Flask=1.0.4 in /home/ec2-user/.local/lib/python3.8/site-packages (from flask-mysqldb) (3.0.0)
Requirement already satisfied: sqlalchemy=1.2.8 in /home/ec2-user/.local/lib/python3.8/site-packages (from flask-mysqldb) (2.2.0)
Requirement already satisfied: Jinja2=3.1.2 in /home/ec2-user/.local/lib/python3.8/site-packages (from Flask=1.0.4->flask-mysqldb) (3.1.2)
Requirement already satisfied: Werkzeug=3.0.0 in /home/ec2-user/.local/lib/python3.8/site-packages (from Flask=1.0.4->flask-mysqldb) (3.0.0)
Requirement already satisfied: blinker=1.6.2 in /home/ec2-user/.local/lib/python3.8/site-packages (from Flask=1.0.4->flask-mysqldb) (1.7.0)
Requirement already satisfied: itsdangerous=2.1.2 in /home/ec2-user/.local/lib/python3.8/site-packages (from Flask=1.0.4->flask-mysqldb) (2.1.2)
Requirement already satisfied: importlib-metadata=3.6.0 in /home/ec2-user/.local/lib/python3.8/site-packages (from Flask=1.0.4->flask-mysqldb) (6.8.0)
Requirement already satisfied: click=8.1.2 in /home/ec2-user/.local/lib/python3.8/site-packages (from Flask=1.0.4->flask-mysqldb) (8.1.7)
Requirement already satisfied: zipp=0.5 in /home/ec2-user/.local/lib/python3.8/site-packages (from importlib-metadata=3.6.0->Flask=1.0.4->flask-mysqldb) (3.17.0)
Requirement already satisfied: MarkupSafe=2.0 in /home/ec2-user/.local/lib/python3.8/site-packages (from Jinja2=3.1.2->Flask=1.0.4->flask-mysqldb) (2.1.3)

```

Figure 10: Code snippet for installing Flask-mysqldb

```

ec2-user~/environment $ pip install --upgrade pip
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pip in /usr/lib/python3.8/site-packages (21.0.1)
Collecting pip
  Downloading pip-23.3.1-py3-none-any.whl (2.1 MB)
    |#####| 2.1 MB 10.4 MB/s
Installing collected packages: pip
Successfully installed pip-23.3.1

```

Figure 11: sudo yum Upgrade


```

29 from gensim.models import Word2Vec
30 import nltk
31 import numpy as np
32 import csv
33 import time
34 import datetime
35
36 from sklearn.preprocessing import StandardScaler
37
38 import tensorflow as tf
39 from tensorflow.keras

```

```

bash -p-172-01-1-167@x Final_Update_LL/Man.py -x
-----
Requirement already satisfied: werkzeug<=1.8.1 in /home/ec2-user/.local/lib/python3.8/site-packages (from tensorflow==2.13.1) (1.0.1)
Requirement already satisfied: cachetools<=4.0,>=2.0.0 in /home/ec2-user/.local/lib/python3.8/site-packages (from google-auth3,>=1.6.3->tensorflow==2.13.1) (4.2.1)
Requirement already satisfied: rsa<=4,>=3.1.4 in /home/ec2-user/.local/lib/python3.8/site-packages (from google-auth3,>=1.6.3->tensorflow==2.13.1) (4.9)
Requirement already satisfied: requests-oauthlib<=0.7.0 in /home/ec2-user/.local/lib/python3.8/site-packages (from google-auth-oauthlib1,>=0.5->tensorflow==2.13.1) (1.3.1)
Requirement already satisfied: importlib-metadata<=4.4 in /home/ec2-user/.local/lib/python3.8/site-packages (from markdown==2.6.8->tensorflow==2.13.1) (4.2.0)
Requirement already satisfied: charset-normalizer<=2 in /home/ec2-user/.local/lib/python3.8/site-packages (from requests3,>=2.21.0->tensorflow==2.13.1) (3.3.2)
Requirement already satisfied: idna<=2.5 in /home/ec2-user/.local/lib/python3.8/site-packages (from requests3,>=2.21.0->tensorflow==2.13.1) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.8/site-packages (from requests3,>=2.21.0->tensorflow==2.13.1) (1.26.3)
Requirement already satisfied: certifi<=2017.4.17 in /home/ec2-user/.local/lib/python3.8/site-packages (from requests3,>=2.21.0->tensorflow==2.13.1) (2023.11.17)
Requirement already satisfied: MarkupSafe<=2.1.1 in /home/ec2-user/.local/lib/python3.8/site-packages (from werkzeug==1.0.1->tensorflow==2.13.1) (2.1.3)
Requirement already satisfied: zipp<=0.5 in /home/ec2-user/.local/lib/python3.8/site-packages (from importlib-metadata==4.4->markdown==2.6.8->tensorflow==2.13.1) (0.5.1)
Requirement already satisfied: pyasn1<=0.6.0,>=0.4.6 in /home/ec2-user/.local/lib/python3.8/site-packages (from pyasn1-modules==0.2.1->google-auth3,>=1.6.3->tensorflow==2.13.1) (0.5.1)
Requirement already satisfied: oauthlib<=3.0.0 in /home/ec2-user/.local/lib/python3.8/site-packages (from requests-oauthlib==0.7.0->google-auth-oauthlib1,>=0.5->tensorflow==2.13.1) (3.2.2)
ec2-user:~/Final_Update_LL/Man.py $

```

Figure 12: Code snippet for installing TensorFlow

XAMPP is widely used in projects for its simplicity and versatility. It provides an integrated environment with Apache for web hosting, MySQL for database management, PHP for server-side scripting, and more. This all-in-one package streamlines the setup of a local development server, enabling developers to create, test, and debug web applications seamlessly.

Apache Friends MENU

XAMPP Apache + MariaDB + PHP + Perl

Welcome to XAMPP for OS X 8.0.28

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the [FAQs](#) section or check the [HOW-TO Guides](#) for getting started with PHP applications.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others.

Start the XAMPP Control Panel to check the server status.

Community

XAMPP has been around for more than 10 years – there is a huge community behind it. You can get involved by joining our [Forums](#), liking us on [Facebook](#), or following our exploits on [Twitter](#).

Figure 13: Code snippet for installing TensorFlow

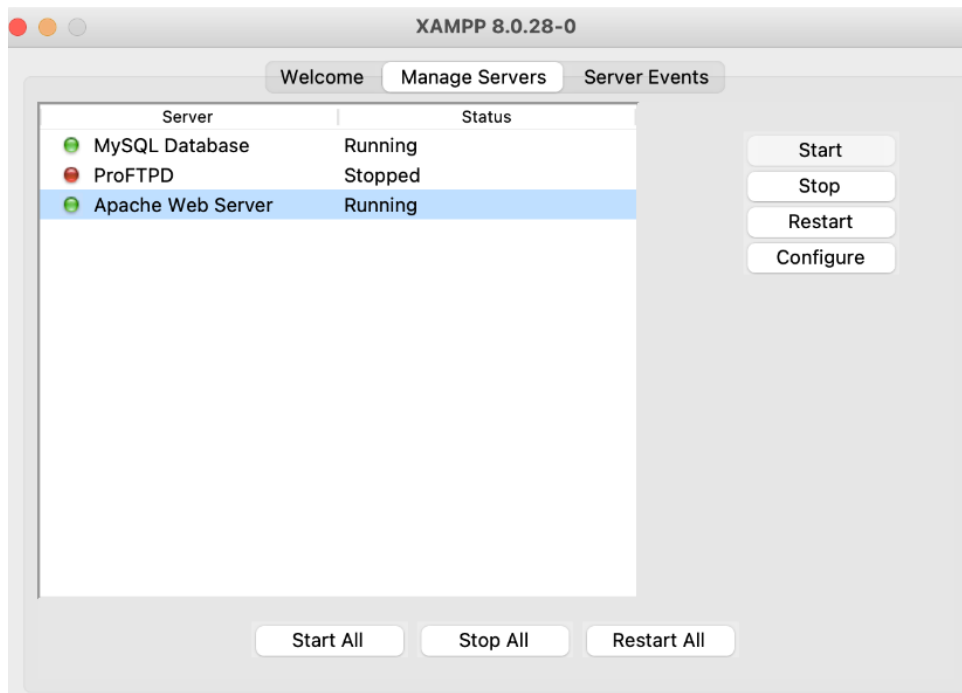


Figure 14: Code snippet for installing TensorFlow

5 Presentation and Demo Video Link

Click on the following link to watch the video: <https://youtu.be/pSSNWkzr9pE>