An Online NLTK Demo

Dr Joel Azzopardi

Dr Claudia Borg

December 2020

ARI 1102

Karsten Guenther

0295697M

Contents

Ubuntu Server Setup

The installation of the Linux Server Image on Virtual Box was completed as shown in the images below.

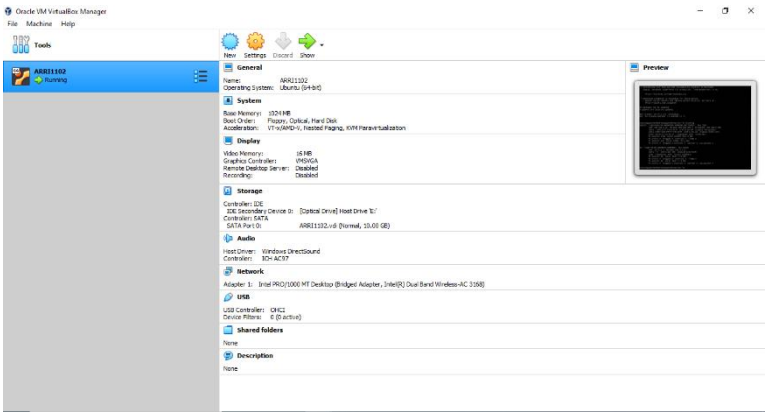


Figure 1

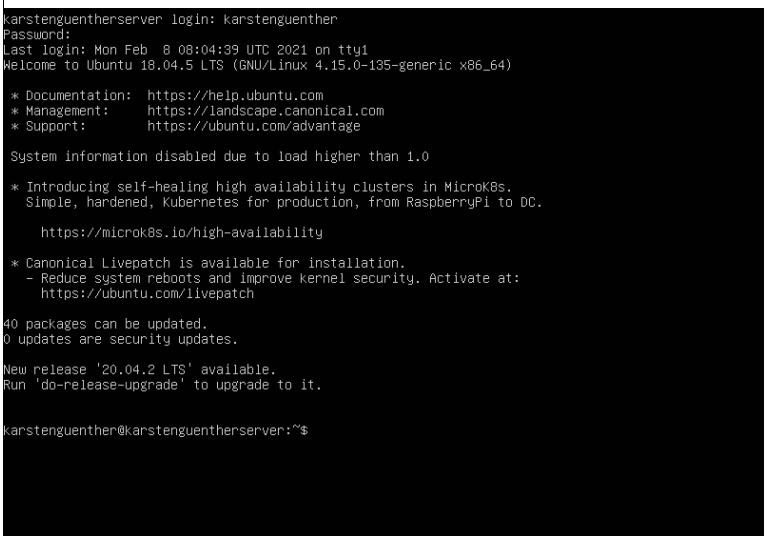


Figure 2

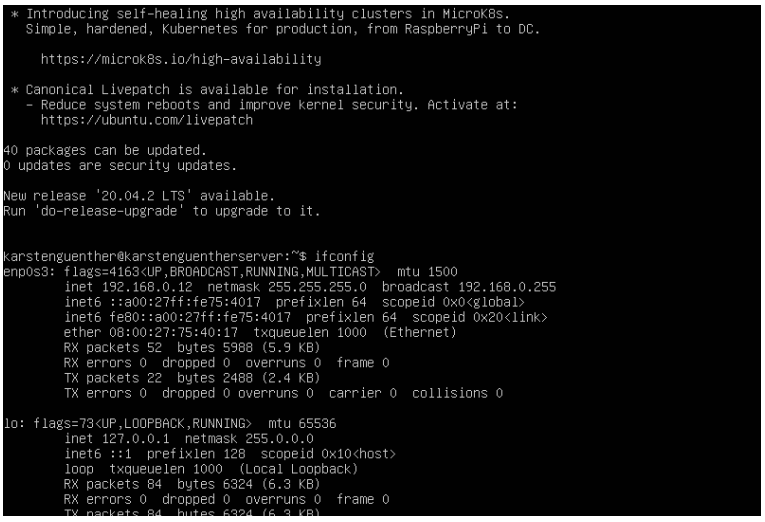


Figure 3

Installation and configuration of web server (Apache 2).

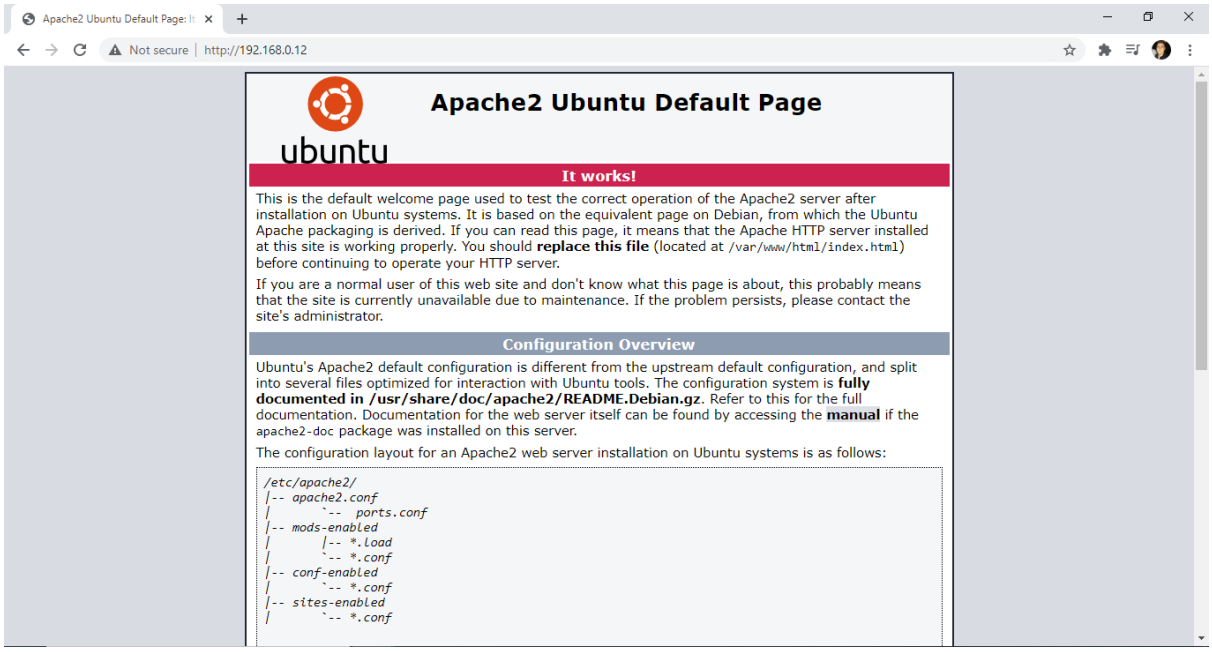


Figure 4

When attempting to transfer the HTML, Json and Python Script files to their respective files through File Zilla, I encountered some permission restrictions that hindered me from completing the file transfer. The problem was not resolved.

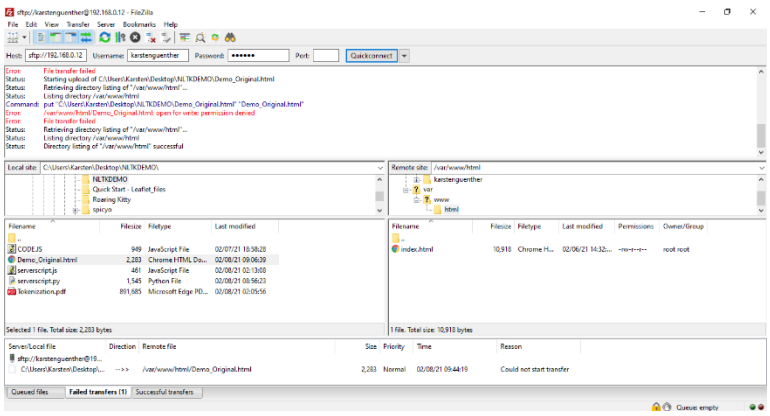


Figure 5

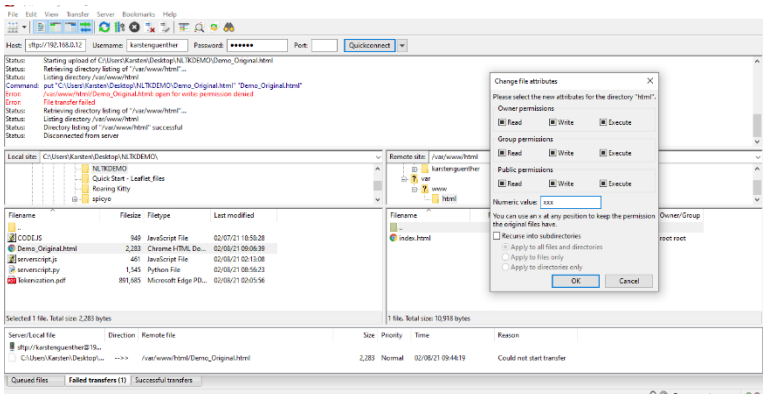


Figure 6

Web Interface

The web interface which is intended to be used by the user to enter a string of text which in turn would present the user with a language check, tokenisation of sentences and words, part-of-speech tagging and named entity recognition. The output of these features was not completed since the connection between the HTML, Json and Python Script files was not established and the transfer of the files on to the virtual machine was not successful. The web server was to be configured so that the web interface communicates with the python script running on the server and returns processed data when data is inputted.

The interface was adapted from simple forms created on w3schools.

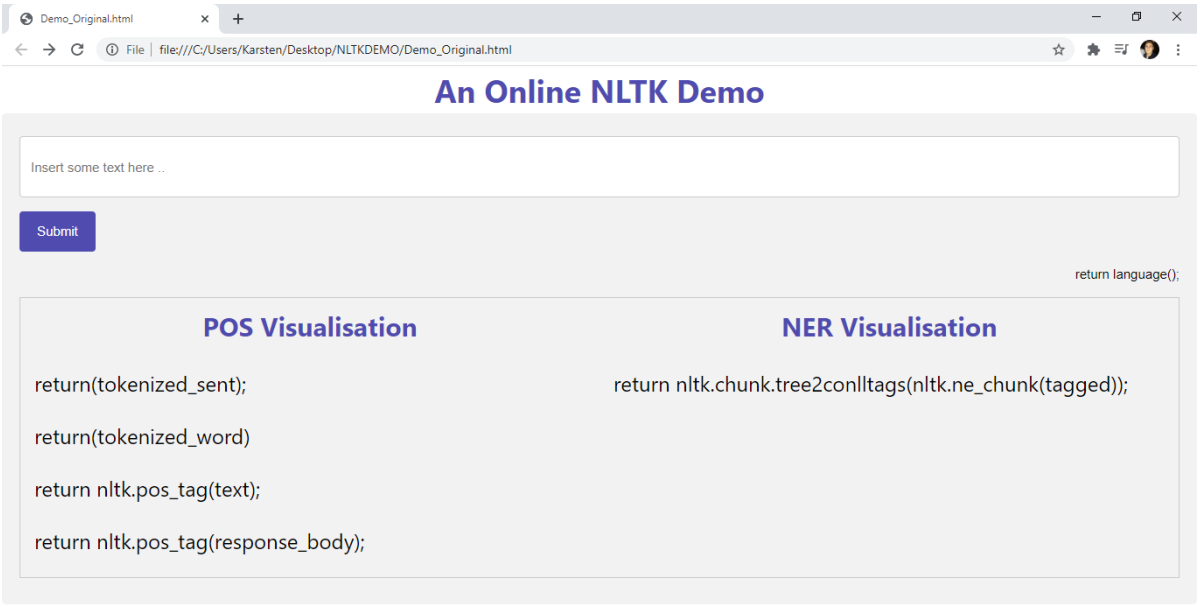
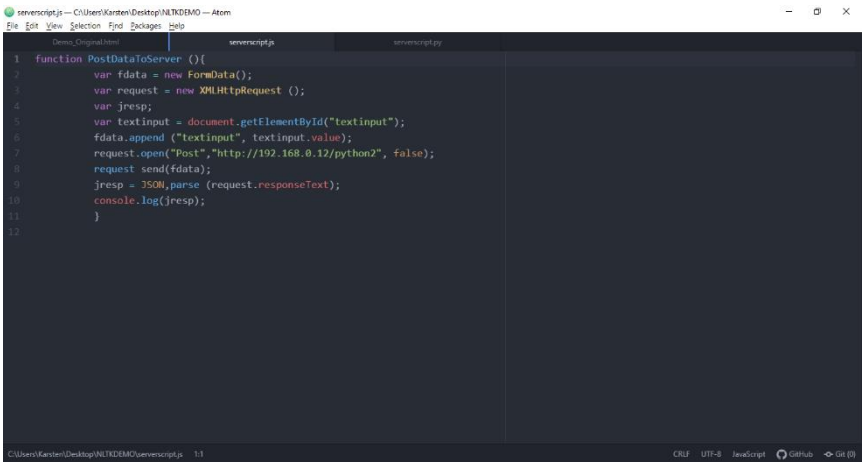


Figure 7

Python server scripting.

The python server scripting failed at returning the required features to the user on the web interface even though the python code works successfully on Jupyter Notebook IDE. The python script from a text editor is presented below and a pdf containing the executed code on Jupyter Notebook is presented in the zip file submitted for reference. The file is called NLTK.pdf.



```
function PostDataToServer (){
        var fdata = new FormData();
        var request = new XMLHttpRequest ();
        var jresp;
        var textinput = document.getElementById("textinput");
        fdata.append ("textinput", textinput.value);
        request.open("Post","http://192.168.0.12/python2", false);
        request send(fdata);
        jresp = JSON.parse (request.responseText);
        console.log(jresp);
        }
```

Figure 8

```
#serverscript to process nltk
import nltk;
import cgi;
import json;
from nltk.tokenize import sent_tokenize;
from nltk.tokenize import word_tokenize;

def getPostDataAsJson (environ):
    post_json = {};
    storage = cgi.FieldStorage(fp = environ['wsgi.input'], environ = environ,
    keep_blank_values = True();
        for k in storage.keys():
            post_json[k] = storage.getvalue(k);
            return (post_json);

def application(environ , start_response):
    form_params = getPostDataAsJson (environ);
    conttype = 'application/json';
    response_body = [json.dumps(form_params)];
    content_length = sum([len(s) for s in response_body]);
    status = '200 OK'
    response_headers = [('Content - Type', conttype ),('Content - Length',str (content_length))];
    start_response ( status , response_headers )
    return response_body;
```

Figure 9

```
def langdete():
    language = detect(response_body))
    return language();

def TokenizedSentence():
    tokenized_sent=sent_tokenize(response_body)
    return(tokenized_sent);

def TokenizedWord():
    tokenized_word=word_tokenize(response_body)
    return(tokenized_word);

def Tagging():
    text = word_tokenize(response_body)
    return nltk.pos_tag(text);

def nltkuniversal():
    return nltk.pos_tag(response_body,tagset='universal'))
    return nltk.pos_tag(response_body));

def NamedEntityRecognition ():
    tagged = nltk.pos_tag(response_body)
    return nltk.chunk.tree2conlltags(nltk.ne_chunk(tagged));
```

Figure 10